Sentiment Analysis of Twitter Data

THESIS

Evan L. Munson, Captain, USA

AFIT-ENS-MS-18-M-148

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this document are those of the author and do not reflect the official policy or position of the United States Air Force, the United States Army, the United States Department of Defense or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENS-MS-18-M-148

SENTIMENT ANALYSIS OF TWITTER DATA

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Master of Science in Operations Research

Evan L. Munson, B.S., M.S.

Captain, USA

22 March 2018

AFIT-ENS-MS-18-M-148

SENTIMENT ANALYSIS OF TWITTER DATA

THESIS

Evan L. Munson, B.S., M.S.
Captain, USA

Committee Membership:

LTC C. M. Smith, Ph.D.
Chair

Dr. B. C. Boehmke
Member

AFIT-ENS-MS-18-M-148

Abstract

The rapid expansion and acceptance of social media has opened doors into users opinions and perceptions that were never as accessible as they are with today's prevalence of mobile technology. Harvested data, analyzed for opinions and sentiment can provide powerful insight into a population. This research utilizes Twitter data due to its widespread global use, in order to examine the sentiment associated with tweets. An approach utilizing Twitter #hashtags and Latent Dirichlet Allocation topic modeling were utilized to differentiate between tweet topics. A lexicographical dictionary was then utilized to classify sentiment. This method provides a framework for an analyst to ingest Twitter data, conduct an analysis and provide insight into the sentiment contained within the data.

AFIT-ENS-MS-18-M-148

To my amazing Wife and Family, who supported me through this entire endeavor. I love and appreciate you more than I can ever express in words

To my God, who through him all things possible.

-ELM

Acknowledgements

I would like to express my gratitude to my faculty research advisor, LTC Christopher Smith, Ph.D., for all of his guidance, patience and assistance.

I would like to thank all of my peers and friends that have assisted me through the coursework and analysis needed for this analysis.

Evan L. Munson

Table of Contents

Page

Page

List of Figures

Figure Page

List of Tables

SENTIMENT ANALYSIS OF TWITTER DATA

I. Introduction

> "We Own the Data." Much like the Army owns the night and thus a key advantage in the physical domains, we must also own the data to gain a competitive advantage in the cyber domain [7].
>
> –John W. Baker
> Major General, USA
> Commanding General, NETCOM

1.1 Background

Recent years have witnessed the rapid growth of social media platforms in which user's can publish their individual thoughts and opinions (e.g., Facebook, Twitter, Google+ and several blogs). The rise in popularity of social media has changed the world wide web from a static repository to a dynamic forum for anyone to voice their opinion across the globe. This new dimension of *User Generated Content* opens up a new and exciting world of insight to individuals, groups, companies, governments, etc. [8].

Social network sites or platforms are defined as web-based services that allow individuals to:

1. Construct a public or semi-public profile within a bounded system.
2. Articulate a list of other users with whom they share a connection.
3. View and traverse their list of connections and those made by others within the system.

The nature and nomenclature of these connections may vary from site to site [9].

As an initial reference point for readers, Facebook was originally a novelty item launched in early 2004 for Harvard-only college students with a @harvard.edu email address. As Facebook's popularity expanded, Facebook began supporting other schools. The new users' were required to have a university email address associated with their institution, this requirement kept the site relatively closed and contributed to users' perceptions of the site as an intimate, private community [9]. Fast forward to the present and anyone can create a Facebook account, which has the impact of intertwining Facebook and other social media sites into nearly every aspect of daily life. The newly constructed social media networks that have arisen from this technology have transitioned from simple text and image sharing to global platforms capable of transmitting celebrity comments across the globe and even social media has become the primary news sources for some individuals.

For the purpose of this research, Twitter will be the focus of our analysis. According to the Pew Research Center, roughly one-quarter of online adults use Twitter [10]. This results in approximately 313 million monthly active Twitter user's across the globe [11]. Twitter is a free, real-time messaging service that is characterized by its 280-character message limit (which was increased from 140-characters in November 2017). Even with its 280-character limit, Twitter has experienced significant growth. For example, Dell has successfully been able to use Twitter to inform its customers of upcoming product discounts [12]. Furthermore, many marketers appreciate Twitter's business value, because it enables companies to easily determine what consumers are saying about their products [12].

1.2 Problem Statement

This research is focused on utilizing Twitter data due to its widespread global acceptance. The rapid expansion and acceptance of social media has opened doors into opinions and perceptions that were never as accessible as they are with today's prevalence of mobile technology. Harvested Twitter data, analyzed for opinions and sentiment can provide powerful insight into a population. This insight can assist companies by letting them better understand their target population. The knowledge gained can also enable governments to better understand a population so they can make more informed decisions for that population. During the course of this research, data was acquired through the Public Twitter Application Programming Interface (API), to obtain tweets as the foundation of data and will build a methodology utilizing a topic modeling and lexicographical approach to analyze the sentiment and opinions of text in English to determine a general sentiment such as positive or negative. The more people express themselves on social media, this application can be used to gauge the general feeling of people.

1.3 Approach

First, an algorithm in the coding language *R* was developed to acquire data through the Public Twitter API. A number of different #hashtags from two different general topics were acquired in order to provide a robust dataset of positive, negative and neutral sentiments.

1.4 Research Objectives

The sentiment analysis of Twitter data will be conducted using the analytic cycle. The analytic cycle is comprised of the following six items: Import, Tidy, Transform, Model, Visualize and Communicate. This framework will guide the research process through the analysis process [13]. During the course of this analysis the following items were developed:

1. Build upon existing text mining methodologies to account for the inherently messy nature of Twitter tweets at input. Tweets can be a mixture of misspellings, acronyms, emoticons, links, images, and tags. The algorithm should be capable of sorting through this amalgam of information and producing usable information.
2. A useful and intuitive manner in which to display and visualize the data for immediate use by analysts and decision makers.
3. An *R* Package for broad accessibility of Sentiment Analysis given a properly formatted text document comprised of comments from social media.

1.5 Assumptions

Twitter results in inherently messy data when acquired through the API. Tweets include "handles" or user-names which can appear as: @johnsmith. Tweets can include any number of #hashtags and other items. Hashtags are dynamic user-generated tags that allow easy access to content for others to easily reference or continue a certain theme[1]. During the analysis, the symbols "@" and "#" were removed while maintaining the handle or hashtag word or phrase, in order to retain that textural information. Additionally stop words were purged utilizing standard tools within the *tidytext R* package. This also assumes the cleaning will not alter the existing sentiment.

[1] `https://www.hashtags.org/featured/what-characters-can-a-hashtag-include/`

Within Twitter, the "RT" word combination will appear on occasion, which marks a tweet as a re-tweet. A re-tweet is when a message from another user has been re-posted onto another Twitter feed. Re-tweeted messages were included in the analysis because they show that other people agreed with or desired to share the original message, and therefore the associated sentiment would be counted again. However, if multiple messages or re-tweets came from the same user and had the same exact time-stamp, the message was not included in order to prevent the inclusion of possible "bot" messages (messages sent by an automated computer program instead of a real user) and inadvertent double counting.

1.6 Summary

This research provides analysts with a process in which Twitter data can rapidly be analyzed to determine the sentiment. The developed process will allow for the sentiment of an individual tweet to be selected and analyzed. The process will allow for a unique individual to be analyzed. In addition, the analysis can be applied across collected #hashtags and determined latent topics within the twitter data. This application provides for the swift determination and visualization of sentiment with the purpose of providing insight into the opinions and thoughts of Twitter users'.

II. Literature Review

Twitter Sentiment Analysis (TSA) tackles the problem of analyzing the messages posted on Twitter in terms of the sentiments they express. Twitter is a novel domain for sentiment analysis (SA) and very challenging. One of the main challenges is the length limitation, according to which tweets can be up to 140 characters. In addition, the short length and the informal type of the medium have caused the emergence of textual informalities that are extensively encountered in Twitter. Thus, methods proposed for TSA should take into account these unique characteristics.

–Anastasia Giachanou and Fabio Crestani [8]

2.1 Overview

Text Mining (TM), Sentiment Analysis (SA), and Opinion Mining (OM) are an interesting method in which to gain an understanding of an individual's general feeling about something of interest. One can think of sentiment analysis as a method that uses technology and algorithms to collect and analyze opinions. "In short, sentiment analysis seeks to highlight what people mean, not just what they say. [14]" One possible use of SA is to leverage Government resources to improve services and communication with citizens. This can be especially useful to certain populations that previously were difficult to reach and underrepresented [15]. The Internet and the Web have now made it possible to observe the sentiments and experiences of those in the vast pool of people that are neither our personal acquaintances or well-known professional critics. Anyone with a computer can voice their opinion to the world on the Internet making their opinions available to strangers and everyone else on the Internet [16]. This presents a unique opportunity for anyone with the knowledge and expertise to extract and analyze these freely disseminated sentiments. "Monitoring these patterns and themes over time could provide officials with insights into the perceptions and mood of an individual or community that cannot be collected through traditional methods (e.g., phone or mail surveys) due to a variety of reasons, including the prohibitive cost and limited reach of traditional methods as well as the limited window of opportunity for influencing or mitigating events as they evolve." [17].

Data mining and subsequent SA and OM of diverse real-time feeds of social streams related to real-world events can be applied to make sense of the vast amount of information generated. By effectively accomplishing this the government could act more effectively on matters both routine (e.g., ongoing issues of public concern) and critical (e.g., major weather or traffic disruption, public safety or rapid response) [17].

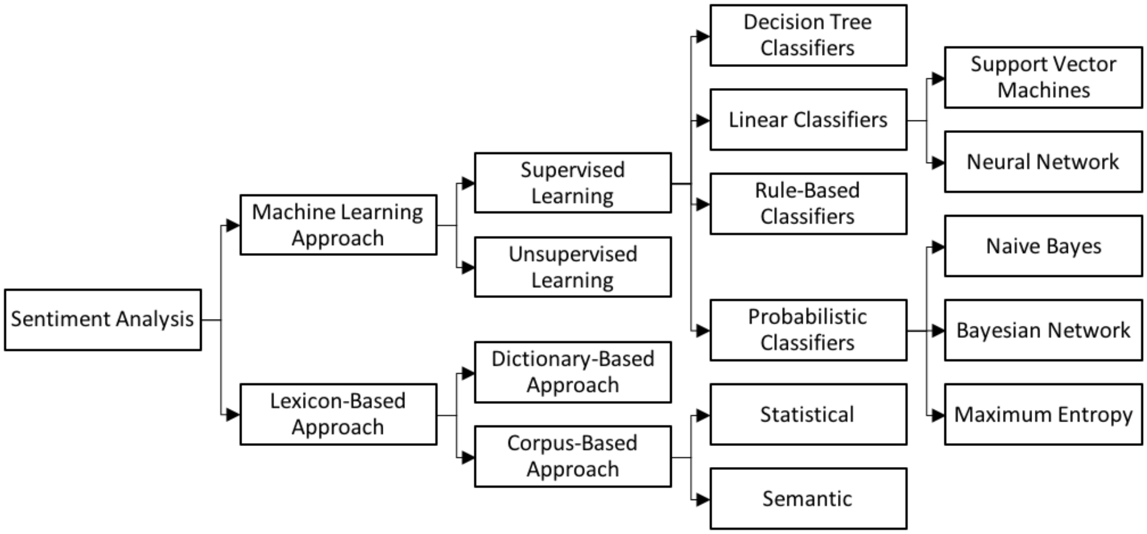


Figure 1. Sentiment Classification Techniques [1]

There are multiple different sentiment classification techniques, as shown in Figure 1. This paper will investigate sentiment analysis utilizing a lexicographical dictionary approach using the SentiWordNet, NRC, Bing and AFINN dictionaries and introduce an Emoji dictionary. Additionally, this research will investigate the use of Latent Dirichlet Allocation (LDA) topic modeling to discover the latent topics associated with the tweet data. The use of LDA is very critical of the number of clusters selected. In order to determine an "optimal" number of clusters four different methods were utilized using a: Markov Chain Monte Carlo algorithm, Density-based method, KL-Divergence, and an Unsupervised Latent Concept Modeling method.

2.1.1 Validation

> *Events like the so-called "Arab Spring" underline the need for the United States and its allies to reliably monitor the global information environment, so that they can build sociocultural understanding, anticipate change before it happens, and plan for appropriate action regarding adversaries and general populations.* [18]

In the 20th century, significant advances were made with regards to radar, sonar, and infrared sensing systems that greatly improved the ability to remotely sense objects through various mediums. However, in the 21st century, the paradigm has shifted to one that now emphasizes cultural understanding with both allied and adversarial governments. Which includes understanding how local populaces could have a critical effect on strategic success or failure for a nation. Ever increasingly it is important to focus and understand human geography, human terrain, and sociocultural analysis [18].

Conventional reconnaissance methods of nations include utilizing different Intelligence, Surveillance and Reconnaissance (ISR) and Indications and Warnings (I&W) methods which are only able to detect physical objects and movements. Future success will require analysts to anticipate how cultures, societies, groups, religion, and ideology influence a population. This will require a capability at the tactical, operational, and strategic levels [18].

One possible method to capture the populace insight is the *Social Radar* as proposed in the MITRE Corporations, Social Radar report [18].

- Rapidly determine sentiment
- Detect and gauge the spread of groups, networks, messages of interest, and sentiment through time and space?

The idea of the social radar is to attempt to capture the requisite societal cues and analyze it in a manner that will support nations similar to the 20th-century advances in aircraft radar.

Twitter Sentiment Analysis could provide an aspect of the social radar with its ability to remotely gather information from a population. Using the *Arab Spring* revolution as an example, conducting a thorough sentiment analysis could be used in a manner to determine the mood and temperance of a protest and determine which kinds of sentiments are being expressed. Which could be ultimately used to determine if a protest (in this example) will become more widespread or remain localized [19].

For example, Twitter #hashtags could be utilized to indicate attitudinal changes in opinions towards more tolerant or intolerant opinions. Furthermore, the use of the analysis could be significantly increased when the identity and location of the user could be determined. For example, it could be critically important to understand whether radical sentiments are originating from Twitter user's in Baghdad or Arizona [20].

Previously Twitter sentiment analysis has been compared to a social radar, another way to think about sentiment analysis is to think of it as a *social barometer* [14]. The social barometer receives unfiltered feedback. With this feedback, an organization can observe the feedback and adjust accordingly [14]. In a business setting, sentiment analysis allows a company to listen to all consumers and not just the most vocal and understand themselves better as a company. Consumers have always talked about products, now however with the commonplace use of social media, organizations can listen and adapt [14].

While sentiment analysis is an impressive tool, it is not without its faults. Human language is a large and complex entity, training or creating an algorithm that is 100% accurate every time is extremely challenging, and the technology is not yet fully there. Furthermore, distinguishing a texts context, tone, skepticism, sarcasm, etc. is a very challenging area of research [14].

Even with multiple challenges and areas of improvement needed for sentiment analysis, it does have a large amount of growth for the foreseeable future. Companies and media organizations are working to develop ways to mine social media and Twitter data. Companies like: Spredfast[1], Microsoft Azure[2], and Lexalytics[3] all provide sentiment analysis services for organizations [21]. Social media is continuing to expand, and with it so will the capability of sentiment analysis. When companies as large as Microsoft begin developing and offering sentiment analysis capabilities, it is a sure sign the product is valuable and will continue to see significant growth in the future.

2.2 Social Media

Social media has experienced significant growth over the past 5 years. Social media networks allow people to share ideas, comments, and digital media across the globe. A subset of social media is micro-blogging services. Compared to conventional blogs in which people can share as much as they want, micro-blogs are much smaller. For example, Twitter, Tumblr, FourSquare, Google+, and LinkedIn are examples of micro-blogs [8]. In Twitter's case, a person can only create a message 280-characters or less. However this “limitation” or “feature” has not dampened Twitter popularity as the service has approximately 328 million monthly active users' who post 500

[1] https://www.spredfast.com/

[2] https://azure.microsoft.com

[3] https://www.lexalytics.com/

million tweets per day [11]. Because of this quantity of daily created data, Twitter is considered to have created one of the largest user generated datasets [8].

2.2.1 Availability

Twitter has a very large and active user base. Twitter has a mobile application for connecting multiple different devices to include both phone, tablet, and desktops for Mac, Windows, Nokia and Blackberry devices. In addition to being able to access Twitter directly from its website[4], Twitter is widely accepted across the globe with the exception of a few countries. Currently China, Iran, and North Korea block access to Twitter for their citizens. Sporadically, Egypt, India, Israel, Pakistan, Russia, South Korea, United Arab Emirates and Venezuela have censored or blocked access or certain tweets in an effort to prevent information sharing [22].

2.2.2 Social Media Benefits

As mentioned above, Twitter has approximately 328 million users' and is a ready dataset for analysis. The nature of Twitter data includes a number of useful variables for analysis. For example, the Twitter API output includes data on when a post was created, how many times it was re-tweeted, and a geo-referenced location (when a user does not disable this option, most disable the geo-referenced option) in addition to the actual tweet from the user.

Through the lens of a military commander, this information is very useful. Analysis of this data could be used to identify individuals or groups who are becoming radicalized, measure their prevalence of support for extremist causes and gauge the extent of support. Geo-referenced tweets can be used to identify how an how an idea, comment, post, etc spread. Additionally, network analysis can be used to counter

[4] `https://Twitter.com/`

the spread of ideas or understand how an idea spreads through social media. Combined together, the information can be used to target individuals, or it can be used to prevent the spread of an undesirable idea [23].

Social media data has been used in the last couple years as a very useful tool to direct actions against hostile elements. For example, a U.S. Air Force unit was able to triangulate an Islamic State in Iraq and the Levant (ISIL) fighters location in 2015. The unit was able to determine the fighter's location and subsequently the ISIL headquarters building, because of a geo-referenced social media post which resulted in a successful bombing sortie [23].

2.2.3 Social Media Challenges

Social media analysis is undoubtedly an important data source. However, it does have its limitations. It is important to remember that any collected social media data is not a representation of an entire population and the data is skewed towards those who participate in social media. Furthermore, social media does not have the same amount of use and popularity in all parts of the world [23].

Individuals around the world, including civilian populations, U.S. allies, and U.S. adversaries, use social media platforms to share information and persuade others. The rapid growth of the communication technologies that underpin social media platforms has given non-state adversaries an asymmetric advantage [23].

> Rapid communications evolutions tend to favor small, agile, less bureaucratic organizations that can more quickly leverage technological advancements without having to negotiate lengthy oversight and authorities processes. The US Department of Defenses advantage in material, financial, and technological resources will be effectively negated if it fails to secure a foothold in these emerging communications spaces. Identification of the most promising techniques and technologies is the crucial first step in positioning to establish relevance in a rapidly changing environment [24].

Many U.S. adversaries have been quick to exploit social media. However, a gap within the Department of Defense (DoD) exists, in which they lack the ability to effectively monitor and utilize social media analytic tools to support awareness of the operating environment [23]. However the the U.S. House of Representatives Committee on Armed Services recognizes this capability gap and mandated the Secretary of Defense to examine the demand for such capabilities and to identify any gaps or areas needing clarification in policy, doctrine, training, and technology capabilities so to consider operational missions for social media analytics, such as battlespace awareness, operational security, and sentiment analysis for counter-messaging adversarial narratives [25].

2.2.3.1 Data and Technology Considerations

One major consideration for the DoD is the acquisition of social media analytics technology and data. The concerns include both the cost and trade-off among the DoDs challenging acquisition strategy. For example, commercial off the shelf (COTS) technology may be attractive to the DoD due to sophisticated technology developed from competitive commercial marketplaces. However, the vendor's technology solution may introduce critical mismatches in context and purpose due to the differences in purpose between commercial and military operational needs. For example, sentiment analysis utilized for a commercial business's brand management would be very different from the capability required for the DoD [23].

There are three concerns associated with utilizing social media data for analysis in the DoD according to Marcellino [23].

- *Data and Technology Acquisition Cost:* There are significant costs associated with acquiring new technology. As mentioned above finding a vendor that is able to provide not just sentiment analysis, but sentiment analysis tailored to the

DoD's needs and requirements is challenging. Furthermore, the sheer volume of social media data will make the acquisition of technology difficult and challenge acquisition methods.

- *Scaling Analysis:* Due to the increase of social media data, developing systems that are scalable could prove to be a challenge. There could be a need to have solutions that are capable of "triaging" data so to present human analysts with manageable streams of data.

- *Standards and Sharing:* DoD will require a shared operating picture for effective social media analysis. The DoD will require an enterprise level solution to develop standards with a shared data architecture. This shared architecture will enable agencies to share raw data, analysis results and data visualizations enterprise-wide. Without a standard, there will be no systematic way to test new technologies or methods.

In order to fully utilize the quantity of social media data that is present and the quantity that will be available in the future, the DoD will need to develop an enterprise level infrastructure to account and manage the supply and requirements [23].

2.2.3.2 Data Source Considerations

While social media is a very accessible source of information, it does have a number of potentially large drawbacks or challenges associated with its utilization. Social media use around the world is variable. This variability will cause challenges in its use. Secondly, users' of social media are self-selected, and, thus the data they share is inherently skewed towards the population they participate in [23].

Additionally, developers wanting to access the Twitter API are limited to the number of tweets they can acquire. The Twitter API is broken into 15-minute windows and allows for 15 requests per rate limit window [26].

For example, to show how sparse social media data could be, 1,000 tweets of each of the following seven #hashtags (#job, #Friday, #fail, #icecream, #random, #kitten) were acquired on 27 August 2017, for a total of 7,000 tweets. In Figure 2, only 997 of the original 7,000 tweets were enabled with the geo-reference option selected.

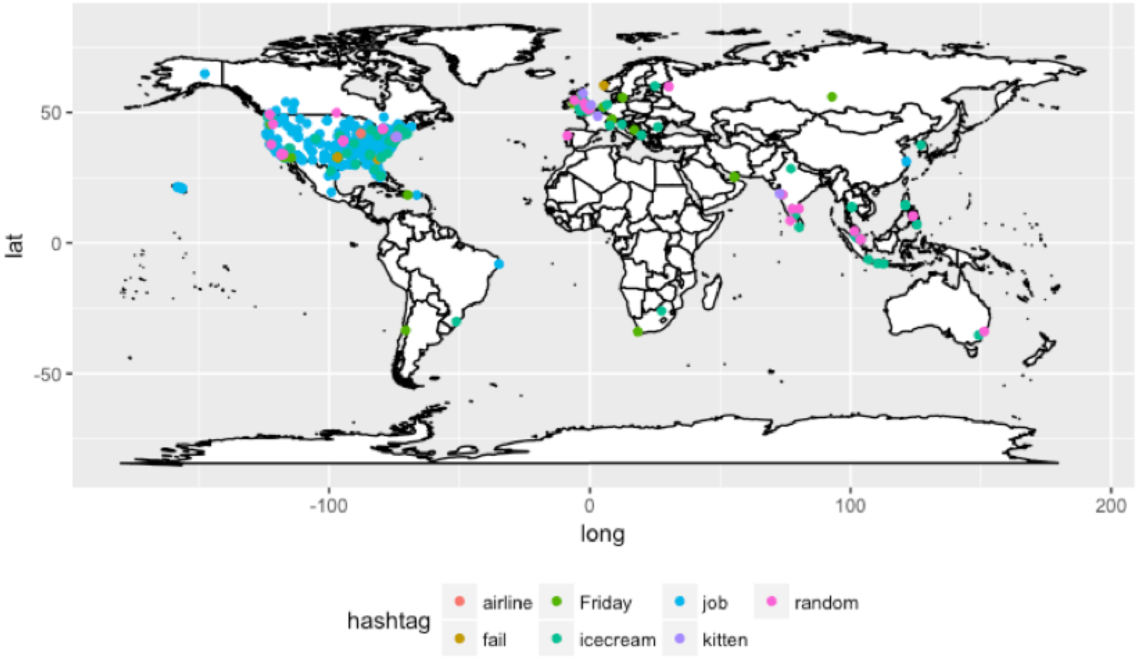


Figure 2. World Map of Tweets

From this map, it can be observed that the United States and Europe generally had a large amount of Twitter use. However, Twitter use in South American, Africa, Russia and China are very small. In these parts of the world, the use of Twitter social media would prove problematic due to the sparsity of tweets in the areas.

2.2.4 Approaches to Utilize Social Media

Previously, some of the challenges and considerations associated with social media have been discussed. Social media analysis is not merely an academic exercise, it is a powerful tool that is currently being used. For example, utilizing social media data approximately 46,000 Islamic State of Iraq and the Levant (ISIL) supporters were identified utilizing Twitter [27]. Identifying members of the network was a three-step process that combined machine approaches for scalability with human-supervised random sample checks for accuracy [23].

The first step was hand sorting a list of known extremist members who are active on Twitter. This proved to be a highly labor-intensive process taking two experts in the order of months to accomplish. From this search, the researchers found 424 ISIL members.

The second step was to use network connections to identify other possible supporters. The researchers utilized a method that depended on the connections that a user may have. The researchers did not look at the users' that followed extremist members but instead looked at the people extremist members followed. Thus in this extremist network example, ignoring those who follow the identified members and instead identifying those being followed by the identified members provides a potentially more accurate picture of the likely network members. After this pass, the researchers identified approximately 43000 members.

The third step in identifying active ISIL network members was to sort the remaining 43,000 by their engagement with ISIL on Twitter as well as their degree of cliquishness and in-network focus. Where:

- Cliques are substructures within a network in which every node is connected to every other node.

- In-network focus refers to the tendency to have more in-network connections than out-of-network connections (interactions with members outside a group).

The researchers were able to sort the 43,000 accounts utilizing cliquishness and in-network focus to provide a much higher classification accuracy than using a single metric. The researchers were able to show that the classification method was $\approx 93\%$ accurate in identifying possible ISIL supporters. This resulted in the classification of $\approx 20,000$ active ISIL supporters [23].

Utilizing social media, the DoD will be able to access large amounts of data that will enable a unique and immensely powerful look into members across the globe.

2.3 Text Mining

2.3.1 Definition

Text Mining is the discovery by computer, of new, previously unknown information, by automatically extracting information from different written resources [28].

2.3.2 Application

One of the most common strategies used in text mining is to identify important entities within the text and attempt to show connections among those entities [28].

2.4 Sentiment Analysis

2.4.1 Definition

Sentiment analysis is the computational study of opinions, sentiments, emotions, and attitudes expressed in text towards an entity. Sentiment analysis is comprised of detecting, extracting and classifying opinions, sentiments and attitudes concerning different topics, as expressed in textual input [29].

It is generally assumed that two or perhaps three values *positive, negative, neutral* are enough to classify the sentiment of a text [30]. The methods to classify the values have grown increasingly sophisticated with more powerful computer capacity, however current classification methods continue to consider the individual or local combination of words and match them against a predefined list of words with fixed sentiment values [30].

2.4.2 Application

Sentiment analysis is an application of natural language processing that focuses on identifying expressions that reflect authors opinion-based attitude [30]. SA helps in achieving various goals like observing public mood regarding political movement, market intelligence, the measurement of customer satisfaction, movie sales prediction and many more [29].

The field of sentiment analysis originated from the computer sciences rather than linguists. The Merriam-Webster's dictionary defines *sentiment* as an attitude, though, or judgment prompted by feeling. Whereas *opinion* is defined as a view, judgment, or appraisal formed in the mind about a particular matter. The difference is subtle, but they indicate that an opinion is more of an individual's concrete view about something, and a sentiment is more of a feeling about something. In this paper, the term opinion will refer to the entire concept of the of sentiment, evaluation, or attitude and associated information an individual holds about that opinion. Sentiment will infer the underlying positive, negative or neutral feeling implied by the opinion [31].

2.4.3 Challenges

Twitter Sentiment Analysis (TSA) tackles the problem of analyzing the messages posted on Twitter in terms of the sentiments they express. Twitter is a novel domain for SA and very challenging. Some of the critical challenges of analyzing TSA is according to Giachanou et. al [8] are:

- Text Length: Each message is limited to 280 characters. The short length proves challenging because very few words are retained for classification.
- Topic Relevance: Previous TSA work did not take into account the topic(s) the tweets were aligned with. One method to determine a topic is to use #hashtags, however, they are determined by a user and could or could not be accurate.
- Incorrect English: Twitter is very informal, and as such many people use very bad and incorrect English which make it very difficult to conduct a lexicographical approach for SA. This is because if a word is not spelled correctly it will not be properly accounted for and the analyst may not glean the proper sentiment from the tweet. Additionally, tweets may include slangs, abbreviations, lengthening of words etc which also prove difficult to correctly classify.
- Data Sparsity: Because many tweets have misspellings or words and phrases that are not contained within a lexicon they will not be properly accounted for when sentiment is scored. This results in very sparse data, in which the sentiment is determined by a comparatively few number of words.
- Negation: Detecting sarcasm and double negatives are very challenging because if not properly accounted for the result will be the opposite of the message's true polarity (positive becomes negative or vice versa).
- Stop Words: Stop words generally have to be removed because they do not provide useful information for an analyst. Typically stop words like the, like, is, who, etc have low sentiment discrimination power and are not scored within

a lexicon dictionary, so they need to be removed.

- Multilingual Content: Twitter is a global entity, therefore it stands that tweets can and will be in foreign languages. Therefore, an analysis language will have to be selected and tweets of the same language need to be collected. Another consideration would be one in which a user combines multiple languages into a single tweet.
- Multinodal Content: Tweets are many times loaded with other digital content. Tweets can contain text, images, movies clips, .gif, weblinks, #hashtags, user handles, etc. Sorting through the multiple different types of content is an important cleaning step.

2.4.4 Feature Selection

The majority of SA and TSA methods detect sentiment based on a feature set. The selected features and their combination play an important role in detecting the sentiment of text. In the domain of microblogs, which Twitter falls into, three different classes of textural features can be observed [8]:

- Semantic Features: The most important words for analysis are sentiment words that contain a positive or negative sentiment.
- Syntactic Features: Syntactic features typically include unigrams, bigrams, n-grams, terms frequencies, and a words parts of speech (POS).
- Stylistic Features: These elements typically can be observed as non-standard writing elements. Some examples are emoticons also known as Emoji.

2.4.5 Lexicographical

Different words and phrases convey positive or negative sentiments which are foundational for sentiment analysis. There are generally three primary approaches to compiling sentiment words: *manual, dictionary-based,* and *corpus based* approaches. The manual method is the most time consuming and is generally used as a check on automated methods. A dictionary approach is an obvious method to compile sentiment words because most dictionaries include synonyms and antonyms. Therefore a small number of seed words can be used to bootstrap based on the synonym and antonym words within a dictionary. In practice this would work as follows: A small set of sentiment words (seeds) with know polarity is collected, this list would then be expanded by searching in another online dictionary for additional synonym and antonyms. The newly found words are then added to the seed list and the iteration continues until no additional words can be found [31]. The dictionary approach can be further fine-tuned and expounded upon by applying additionally sophisticated techniques.

2.5 Languages

2.5.1 Internet Languages

The world has a multitude of spoken languages. For the purposes of this analysis, the quantity of the worlds spoken languages is not as important as identifying the most used languages on the Internet. As of June 2017, Table 1 displays the top ten spoken languages of the world. From the table, it can be seen that the top ten languages used on the Internet account for 76.9% and nearly three billion of the worlds Internet Users. This is useful for analysis as the bulk of sentiment analysis can focus on a relatively small number of languages.

Table 1. Top Ten Languages Used on the Internet [6]

Rank	Language	Internet Users	Internet Users % of World Total
1	English	984,703,501	25.3%
2	Chinese	770,797,306	19.8%
3	Spanish	312,069,111	8.0%
4	Arabic	184,631,496	4.8%
5	Portuguese	158,399,082	4.1%
6	Indoneisan / Malaysian	157,580,091	4.1%
7	Japanese	118,453,595	3.0%
8	Russian	109,552,842	2.8%
9	French	108,014,564	2.8%
10	German	84,700,419	2.2%
Top Ten Languages		2,988,902,007	76.9%
All remaining World Languages		896,665,611	23.1%
Total		3,885,567,618	100%

2.5.2 WordNet

> *Because meaningful sentences are composed of meaningful words, any system that hopes to process natural languages as people do must have information about words and their meanings. This information is traditionally provided through dictionaries, and machine-readable dictionaries are now widely available. But dictionary entries evolved for the convenience of human readers, not for machines. WordNet[5] provides a more effective combination of traditional lexicographic information and modern computing.*
>
> –George A. Miller [32]

WordNet is an online lexical database. English nouns, verbs, adjectives, and adverbs are organized into sets of synonyms, each representing a lexicalized concept. Semantic relations link the synonym sets [32].

[5] WordNet is a registered trademark of Princeton University: `http://wordnet.princeton.edu/`

2.5.3 SentiWordNet

SentiWordNet is an enhanced lexical resource explicitly devised for supporting sentiment classification and opinion mining created by Andrea Esuli, and Fabrizio Sebastiani. SentiWordNet has gone through multiple revisions and starting with the introduction and publication of SentiWordNet 1.0 in 2006, SentiWordNet 1.1 briefly mentioned in a technical report in 2007, SentiWordNet discussed in Andrea Esuli Ph.D. thesis in 2008 and finally SentiWordNet 3.0 which was introduced in 2010 [33].

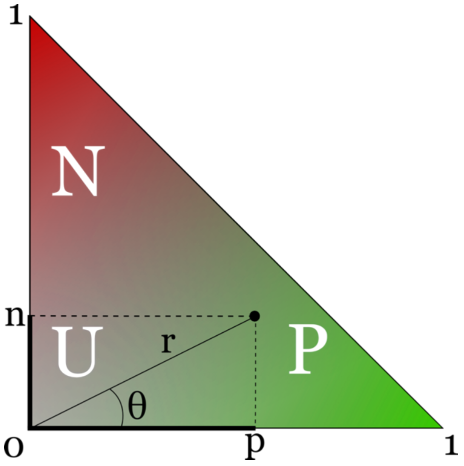


Figure 3. Sentiment Plane [2]

SentiWordNet is the result of automatic annotation of all the synsets[6] of WordNet, according to the notions of *positivity*, *negativity*, and *neutrality*. Each synset is associated with three numerical scores pos, neg and obj which indicate how positive, negative, and objective (neutral) the terms within the synset are. Different senses of the same term can, therefore, have different opinion related properties [32].

SentiWordNet derives the positive, p and negative, n polarity scores assigned to the synsets independently. Such that the sum of positive, negative and neutral scores is 1. Geometrically, a synset is a point in Cartesian space where its x coordinate is its positive score and the y coordinate is the negative score. Where $x + y \leq 1$, therefore

[6] synonyms

the sentiment plane is restricted to a triangle, as seen in Figure 3 [2].

Synsets that lie in the P-region are *positive*, synsets that lie in the N-region are *negative*, and synsets that lie in the U-region are *neutral*. However, as can be seen in Figure 3, these regions are not clearly defined [2].

Utilizing a polar to Cartesian transformation, $\theta = \tan^{-1}(\frac{n}{p})$ and $r = \sqrt{p^2 + n^2}$ where, p = positive score, n = negative score, and r = objective score.

Finally the SentiWordNet Objective Score, or SentiWordNet Sentiment Score $\bar{s}_S$ can be calculated as $\bar{s}_S = 1 - (p - n)$ Where the Sentiment Score has the range: $0 < \bar{s}_S < 1$.

2.5.4 NRC

The NRC lexicon is different than the previously mentioned SentiWordNet. Where the SentiWordNet lexicon assigns a sentiment score, $\bar{s}_S$ the NRC lexicon includes the emotions associated with a word. The NRC Emotion Lexicon, or **EmoLex** accomplishes this by identifying the following emotions: *anger, anticipation, disgust, fear, joy, sadness, surprise,* and *trust*. Additionally, a word is also classified as positive or negative. Often, different emotions are expressed through different words, therefore, a list of emotions and words that are indicative of each emotion is likely to be useful in identifying emotions in text. These emotions were chosen as they are viewed as the basic and prototypical emotions. More complex emotions can be viewed as a combination of these basic emotions. For example, *delightful* and *yummy* indicate the emotion of joy whereas, *gloomy* and *cry* are indicative of sadness [34].

In order to classify these emotions and develop EmoLex, the developers utilized an online resource where a task was submitted to a Mechanical Turk. A Mechanical Turk is a crowd-sourcing platform especially suited for tasks that can be accomplished over the Internet [35]. To utilize this crowd-sourcing platform the requester breaks their desired tasks into small independently solvable units called Human Intelligence Tasks (HITs) where they are uploaded onto the Mechanical Turk website. Individuals that respond to the HITs are considered Turkers and receive compensation for solving each HITs. The first set of annotations completed by the Turkers was completed in about nine days, where the Turkers spent about a minute to answer each question and earned a compensation that worked out to slightly more than $2 [34].

Once the assignments were completed by the Turkers, tasks that were not properly completed were removed. Additionally, the Turkers data was removed if they did not properly answer a word choice question. Furthermore, if a Turker obtained an overall score that was less than 66.67% on the word choice questions, then all task received from that Turker were rejected because it showed that the Turker was not familiar with the words associated with the HITs. By utilizing this crowd-sourcing method the Emotion Lexicon was created.

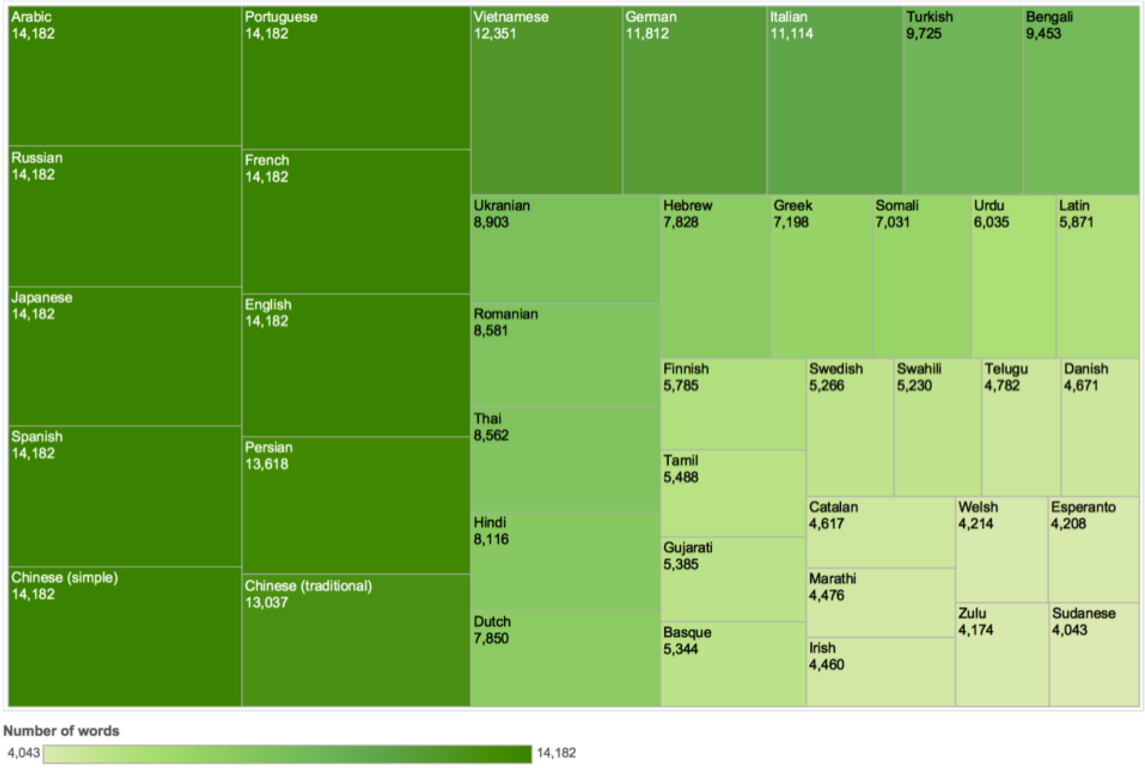


Figure 4. Number of Entries in the NRC Emotion Lexicon, By Language [3]

The original EmoLex was comprised of 14,182 classified English words. To further expand the NRC EmoLex dictionary, the English words were ran through Google Translate in 2015. This expanded EmoLex into a total of 41 languages. Despite some cultural differences, the majority of effective norms are stable across most languages. The EmoLex languages are: English, Arabic, Basque, Bengali, Catalan, Chinese (simplified), Chinese (traditional), Danish, Dutch, Esperanto, Finnish, French, German, Greek, Gujarati, Hebrew, Hindi, Irish, Italian, Japanese, Latin, Marathi, Persian, Portuguese, Romanian, Russian, Somali, Spanish, Sudanese, Swahili, Swedish, Tamil, Telugu, Thai, Turkish, Ukranian, Urdu, Vietnamese, Welsh, Yiddish, and Zulu [3].

As mentioned above, the majority of norms were stable across most languages. In Figure 4 all the different translated languages can be observed with the applicable amount of words for that language that were properly translated. Of note, when Table 1 is compared to Figure 4 it can be observed that eight of the ten (English, Chinese, Spanish, Arabic, Portuguese, Japanese, Russian, and French) most used Internet languages are fully captured by the NRC Emotion Lexicon and the remaining two (Indonesian / Malaysian and German) are generally captured by the NRC Emotion Lexicon.

2.5.5 Bing

The Bing lexicon was created by Dr. Bing Liu and is a list of 6,788 positive and negative English words. Like SentiWordNet it was created to explore sentiment, however, the Bing lexicon was specifically tailored to explore customer product reviews. Bing was based on the work utilizing WordNet. Nouns, verbs, adjectives, and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept [36].

To develop the Bing dictionary, the adjective synonym and antonym set in Word-

Net were used to predict the semantic orientation of adjectives. Adjectives were organized in clusters, where half the cluster was associated with the synonym and the other half the antonym. Other words associated with the synonym and antonym represented the senses that are similar to the primary adjective and completed the cluster. To ensure a broad range of adjectives, 30 Seed adjectives were used to start the dictionary. Once the adjectives orientation was predicted, it was added to the seed list and the analysis continued through the remaining WordNet adjectives to produce the Bing lexicon [36].

2.5.6 AFINN

The AFINN lexicon dictionary was initially developed in 2009 to conduct sentiment analysis on tweets related to the United Nations Climate Conference. One of the goals of this lexicon dictionary was to include Internet slang terms (phrases like: "WTF" and "LOL") and obscene words. The most current version of AFINN contains 2477 unique words and utilizes a scoring range from -5 (very negative) to +5 (very positive). The AFINN dictionary was manually constructed and initiated from a list of obscene words and a few positive words. To build the word list words from the public domain *Original Balanced Affection Word List*[7], Internet slang from Urban Dictionary[8], *The Compass DeRose Guide to Emotion Words*[9], and the Microsoft Web n-gram similarity Web service[10] were used to discover relevant words [37]. These words were combined and generated the AFINN lexicon dictionary.

[7] http://www.pitt.edu/~gsiegle/wordlist/index.htm

[8] http://www.urbandictionary.com

[9] http://www.derose.net/steve/resources/emotionwords/ewords.html

[10] https://azure.microsoft.com/en-us/services/cognitive-services/web-language-model/

2.6 Topic Modeling

Topic modeling is a method to determine the latent topics buried within a text. One method to determine these topics is to utilize Latent Dirichlet Allocation (LDA). The basic construct of LDA is it treats each document as a mixture of topics and each topic as a mixture of words. In this case, the documents are tweets, comprised of multiple words.

LDA is based on two general principles:

- Every document is a mixture of topics
- Every topic is a mixture of words

LDA is a method for estimating both of these constructs at the same time and delivering a mixture of topics that describe each document.

2.6.1 Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a method to discover the topics associated with a body of text. LDA captures the word correlations in the corpus of the document with a low-dimensional set of multinomial distributions called topics. LDA views a document as a distribution over many topics, and a topic is viewed as a distribution of many words [38].

To generate a document, LDA samples a document-specific multinomial distribution over topics from a Dirichlet distribution, then repeatedly samples the words in the document from the corresponding multinomial distributions [38].

A Dirichlet distribution can be thought of as a probability mass function (PMF) that lies in $(k-1)$-dimensional probabilities simplex, which is a surface in $\mathbb{R}^k$ denoted by Δ_k and defined by the set of vectors whose k components are non-negative and sum to 1 [39].

Let $Q = [Q_1, Q_2, \ldots, Q_k]$ be a random PMF, where $Q_i \geq 0$ for $i = 1, 2, \ldots, k$ and $\sum_{i=1}^{k} Q_i = 1$, additionally, suppose $\alpha = [\alpha_1, \alpha_2, \ldots, \alpha_k]$, where $\alpha_i > 0$ for each i, and $\alpha_0 = \sum_{i=1}^{k} \alpha_i$. Q is said to have a Dirichlet distribution with parameter α, denoted by $Q \sim Dir(\alpha)$, if it has $f(q; \alpha) = 0$. If q is not a PMF, and if q is a PMF then,

$$f(q; \alpha) = \frac{\Gamma(\alpha_0)}{\prod_{i=1}^{k} \Gamma(\alpha_i)} \prod_{i=1}^{k} q_i^{\alpha_i - 1}, \tag{1}$$

represents a Dirichlet distribution [39]. The Dirichlet distribution can model the distribution of words in text documents with k possible, words where a document can be represented by a PMF of length k produced by normalizing the empirical frequency of the words. A group of documents produces multiple PMFs and can be fit to a Dirichlet distribution to capture the variability of the PMFs [39].

2.6.2 Number of Topics

One of the challenges of utilizing LDA is determining the correct number of topics that are within the latent structure of the text. Four different LDA methods have been developed to address this challenge. The methods are implemented using the R statistical language and environment for statistical computing [**?**] and accessed using *RStudio*, which is an integrated development environment for R [40]. Within the R language, the *ldatuning* [41] package has been developed to execute all four of these methods these methods:

- Griffiths and Steyvers [42] utilize a Markov Chain Monte Carlo algorithm.
- Cao, Xia, Li, Zhang, and Tang [38] utilize a Density-Based method.
- Arun, Suresh, Veni Madhavan, and Narasimha Murthy [43] utilize a method computed in terms of KL-Divergence.

- Deveaud, SanJuan, and Bellot [44] utilize an Unsupervised Latent Concept Modeling (LCM) method.

2.6.2.1 Markov Chain Monte Carlo Algorithm

One method in which to determine the contribution of different topics within a document is to examine each topic as a probability distribution. Therefore viewing each text document as a probabilistic mixture of topics. If T topics exist, the probability of the ith word in a document is

$$P(w_i) = \sum_{j=1}^{T} P(w_i|z_i = j)P(z_i = j), \tag{2}$$

where, z_i = latent variable indicating the topic from which the ith word was drawn. $P(w_i|z_i = j)$ is the probability of the word w_i under the jth topic, and $P(z_i = j)$ provides the probability of choosing a word from topics j in the current document [42].

Viewing documents as mixtures of probabilistic topics makes determining the topics in a group of documents containing T topics over W unique words as $P(w|z)$ with a set of T multinomial distributions ϕ over the W words, such that $P(w|Z = j) = \phi_w^{(j)}$ and $P(Z)$ with a set of D multinomial distributions θ over the T topics, such that a word in document d, $P(z = j) = \theta_j^{(d)}$ [42].

In order to discover the topics utilizing this approach, ϕ and θ are not explicitly estimated. Instead, the posterior distribution over the assignments of words to topics, $P(z|w)$ is considered. Estimates of ϕ and θ are then obtained by examining the posterior distribution. However, evaluating $P(z|w)$ requires the computation of a probability distribution over a large scale discrete space. To account for this challenge, a Monte Carlo procedure was performed that provided an easy to implement algorithm, with low memory usage, with competitive speed and performance [42].

Finally, this method is a cluster maximization technique.

2.6.2.2 Density-Based Method

The influence of the number of clusters, K selected, can greatly affect LDAs results because of the discrimination between topics. When a too small K is selected, two topics may overlay on a word which results in a close proportion, and a strong correlation between the two topics causes an unstable factor in the topic model. Additionally, if a K is selected that is too large, the words assigned to the topics results in a meaningless result in which the words are forced into nonsensical topics. The sweet spot is when the correct K is chosen which results in little overlap with the words between topics [38].

The density-based method proposes a method by which to measure the distance between topic distributions. The best selection of K is correlated with the distances between topics. The goal of the method is to result in a topic similarity between clusters that will be as great as possible intra-cluster, and as small inter-cluster. A large intra-cluster similarity, shows a cluster to have a more explicit meaning, and a smaller similarity between intra-clusters, indicate a more stable topic structure [38].

The correlation between the topics

$$Corr(T_i, T_j) = \frac{\sum_{v=0}^{V} T_{iv} T_{jv}}{\sqrt{\sum_{v=0}^{V} (T_{iv})^2} \sqrt{\sum_{v=0}^{V} (T_{jv})^2}}, \tag{3}$$

is computed using standard cosine distance [38]. The cosine distance

$$ave\_dis(structure) = \frac{\sum_{i=0}^{K} \sum_{j=i+1}^{K} Corr(T_i, T_j)}{K \times (K-1)/2}, \tag{4}$$

between every pair of topics can also be computed [38]. Where a smaller *ave_dis* infers to a more stable topic structure. Finally, this method is a cluster minimization technique.

2.6.2.3 KL-Divergence Method

The Kullback-Leibler (KL) Divergence method works to discover the optimal number of topics by observing that divergence values are higher for non-optimal numbers of topics which results in a “dip” at the correct number of topics chosen. This method views LDA as a factorization mechanism. A given corpus of text is split into M_1 and M_2. Which are two matrix factors, given by $C_{d*w} = M1_{d*t} \times Q_{t*w}$. Where d is the number of documents, w is the size of vocabulary. The caliber of the split depends on the number of chosen topics, t. The quality of the split is computed using the symmetric KL-Divergence method [43]. Finally, this method is a cluster minimization technique.

2.6.2.4 Unsupervised Latent Concept Modeling

Previously, Arun, Suresh, Veni Madhavan, Narasimha, Murthy, and Cao, Xia, Li, Zhang, Tang, utilized methods that computed similarities between pairs of topics, while varying the number of topics [43, 38]. However, the Unsupervised Latent Concept Modeling method proposes a heuristic to estimate the number of topics by maximizing the divergence D between all pairs (k_i, k_j) of LDA's topics. The number of topics

$$\hat{K} = \text{argmax}_K \frac{1}{K(K-1)} \sum_{(k,k') \in \mathbb{T}_K} D(k \parallel k'), \tag{5}$$

is then estimated [44]. Where K is the number of topics, $\mathbb{T}_K$ is the set of K topics. Finally, this method is a cluster maximization technique.

III. Methodology

3.1 Overview

The methodology used in the report consists of acquiring tweets through the Twitter API; cleaning, tidying and exploring the tweet data; topic modeling; sentiment scoring; summarizing and visualizing results. The overarching flow of the analysis can be observed in Figure 5.

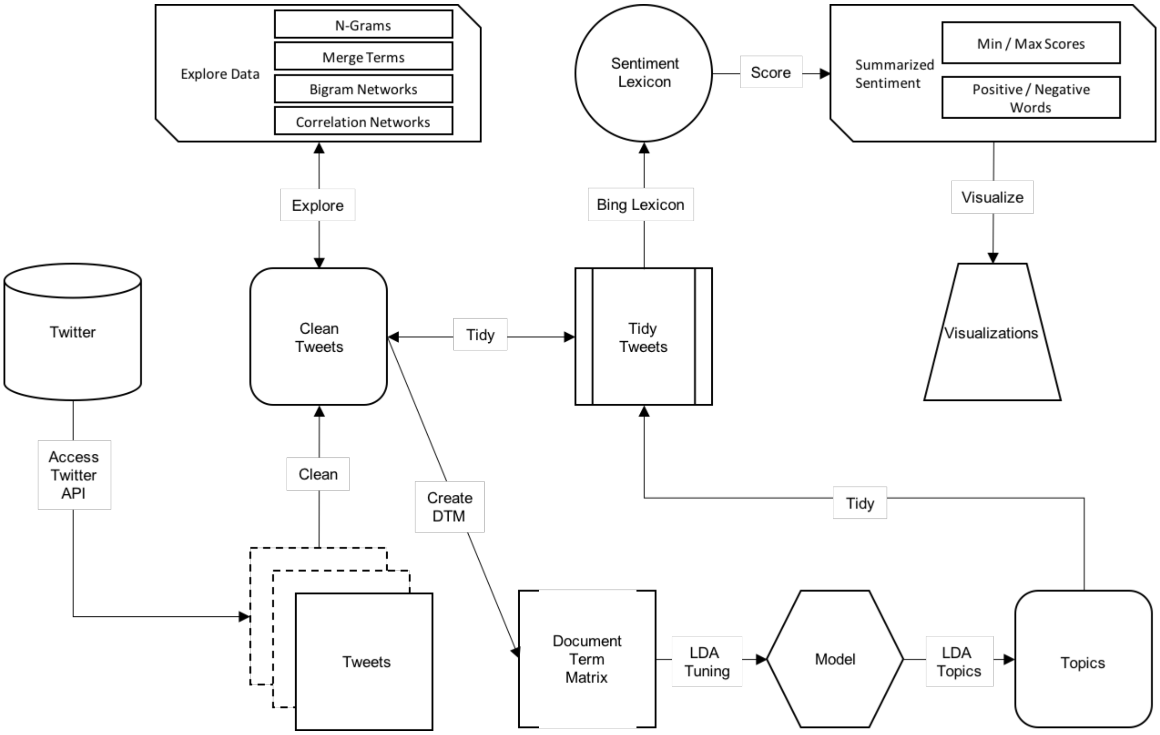


Figure 5. Methodology [4]

3.2 Twitter API

An API is a set of subroutine definitions, protocols, and tools for building application software. In general terms, it is a set of clearly defined methods of communication between various software components.

In this analysis, the actual Twitter data had to be acquired, whereas in other instances a user may already have access to the data. In order to access the Twitter API, the below, intermediate steps must first be completed [45]:

- Create a Twitter[1] account or sign into existing account.
- Using a Twitter account, sign into the Twitter Developers page[2]
    - Navigate to My Applications.
    - Create a new application.
    - Fill out the new application form.
    - Scroll down and click on Create my access token button.
    - Record Twitter access keys and tokens
- Install and load *R* packages.
- Create and Store Twitter Authenticated Credential Object.
- Authorize Twitter application to use the account.
- Extract tweets.

Of note, the Twitter API rate limits the number of requests sent to it. The API allows for 15 calls every 15 minutes [11]. This analysis collected tweets based on #hashtags. The first script requested 1,000 tweets for 11 different #hashtags, without being rate limited because 11 calls were made to the API. Immediately after the first script was run a second script would request 1,000 tweets for 10 different #hashtags, this request would be rate limited and delayed for 15 minutes because the API had received 21 requests within the allotted 15 minute API window.

[1] `https://Twitter.com/`

[2] `https://dev.Twitter.com/apps/`

The resulting Twitter data is returned to the *R* user and contains 16 columns of data per tweet. The different column names and item descriptions can be observed in Table 2

Table 2. Twitter Data Description

Item	Description
text	The text of the status
favorited	Whether this status has been favorited
favoriteCount	Denotes the number of times a tweet has been favorited or liked
replyToSN	ID of the user this was in reply to
created	When this status was created
truncated	Whether this status was truncated
replyToSID	Internal Twitter ID of the tweet the reply was to
id	ID of this status
replyToUID	ID of the user this was in reply to
statusSource	Source user agent for this tweet
screenName	Screen name of the user who posted this status
retweetCount	The number of times this status has been retweeted
isRetweet	TRUE if this is a retweet
retweeted	TRUE if this status has been retweeted
longitude	Twitter georeferenced location for the longitude of tweet
latitude	Twitter georeferenced location for the latitude of tweet

3.3 Cleaning, Tidying, and Exploration

3.3.1 Cleaning

As discussed in section 2.4.3, analyzing Twitter poses numerous challenges. Twitter data is inherently messy as it includes: text, emojis, images, videos, web-links, misspellings, etc. In order to conduct an analysis, the data must be cleaned. The first action when cleaning the data was assigning each tweet a unique key. In this analysis the key is a combination of the user "id" and the "created" date of the tweet. The key is used to remove duplicate tweets and attempt to prevent bot tweets from entering the dataset.

For the purpose of this analysis, a retweet and a duplicate tweet are not necessarily the same. A duplicate tweet is one that has the same exact key as another tweet and could have occurred because a bot was able to simultaneously post the tweet(s). A retweet way be the same tweet shared across multiple different users' and would result in a unique key each instance. Retweets were maintained in the dataset because each shows that multiple users' agreed with or liked the original tweet enough to re-share that tweet again to the community.

For this analysis the following cleaning actions were executed:

- Remove all web-links, as no discernible should be able to be gleaned from the links.
- Remove the "#" symbol from all #hashtags but retain the words associated with the #hashtag.
- Remove the "RT" or retweet identification from all tweets. The actual symbol does not add any useful information. However as mentioned earlier the cumulative effect of multiple retweets was included.
- Remove all punctuation. The punctuation has no value in the sentiment classification.
- Remove all emojis. For the purpose of this analysis, emojis will not be included.
- Remove all stop words. Stop words are words that do not provide any value in sentiment classification. Stop words are common words such as: "the", "of", "to", etc.

3.3.2 Tidying

There are a number of ways to approach sentiment analysis. In the case of this analysis the "tidy approach" was utilized. The tidy approach is an approach that maintains the data in a specific structure.

- Each variable is a column.
- Each observation is a row.
- Each type of observational unit is a table.

For a format to be considered tidy, it should consist of a table with one-token-per-row. Where a token is a unit such as text, words, or some other item to be analyzed using many of the tools and packages that have been designed around the tidy universe. The tidy method was chosen because it brings with it the tools in the *dplyr* package, as much of the analysis, is conducted using a dataframe which is easy to view and can assist an analyst to visualize any code changes [4]. The data in this report was transformed into a tidy structure by each word within a tweet.

3.3.3 Exploration

Once the Twitter data was collected a cleaning, tidying and exploring cycle was conducted. The purpose of this cycle was to explore and prepare the data for further analysis. During this phase the analyst has four primary tools: N-Gram, Merge Terms, Bi-Gram Network, and Correlation Networks.

N-Grams are consecutive sequences of words. A Uni-Gram is a single word, a Bi-Gram is a combination of two consecutive words and Tri-Gram is a combination of three consecutive words [4]. An N-Gram displays the word(s) in question and shows the number of times that combination of word(s) appears within the text. For the purpose of this analysis and the limited amount of text contained within a tweet, only N-Grams up to Tri-Gram were incorporated, as much higher would start returning all the text within a tweet.

Merging terms was developed as a way to prevent redundancy in the analysis. For example, many tweets may refer to the same entity in multiple different ways: President Trump, The U.S. President, POTUS, Trump, President Donald Trump, Donald Trump, etc. While each entry is different, they all refer to the same individual. If they are left as is, multiple different and redundant entries would percolate through an analysis. However, by merging terms the reference becomes one item. For the above example, all the entries could become: "pdjt" which could be an acronym for: President Donald J. Trump, and would remove the redundancy of the previous terms.

Bi-Gram networks build off computed Bi-Grams. Bi-Gram networks serve as a visualization tool that displays the relationships between the words simultaneously as opposed to a tabular display of Bi-Gram words [4]. The Bi-Gram Network plots display the connection between the Bi-Gram words, referred to as nodes, connected by an edge(s) which displays the number of times that word pair is used. In this analysis, the edge varies in width to represent the number of times that combination is observed. A skinny edge width displays a low occurrence and a thick edge width displays high occurrence.

Correlation networks aesthetically look similar to the Bi-Gram network. However, instead of displaying the number of times a word pair is used, the correlation between the words is displayed. In this analysis, the focus will be on the ϕ coefficient, which is equivalent to the Pearson correlation. The ϕ coefficient explains how much more likely it is that both words X and Y appear, or neither do than that one appears without the other [4].

For example, in Table 3, n_{11} is the number of tweets in which word X and Y

Table 3. Correlation [4]

	Has Word Y	No Word Y	Total
Has Word X	n_{11}	n_{10}	n_1
No Word X	n_{01}	n_{00}	n_0
Total	n_1	n_0	n

appear, n_{00} is the number in which neither appear, n_{10} and n_{01} is when one appears without the other. For Table 3 the

$$\phi = \frac{n_{11}n_{00} - n_{10}n_{01}}{\sqrt{n_1 n_0 n_0 n_1}}, \tag{6}$$

can then be calculated [4]. The correlation between words will assist in displaying how often words appear together relative to how often they appear separately.

3.4 Topic Modeling

For the purpose of this report, the optimal number of topics and the LDA algorithm were calculated using previously developed *R* packages.

The optimal number of topics was determined using the *ldatuning* package, which executes the: Markov Chain Monte Carlo, Density-Base, KL-Divergence and unsupervised Latent Concept Modeling methods [41]. The execution of this package returns an LDA tuning plot, and an example can be seen in Figure 6.

The LDA tunning graph in Figure 6 is a two-part display. The upper half of the plot displays the topic minimization using the Density-Based method and KL-Divergence methods. The bottom half of the plot displays the topic maximization using Markov Chain Monte Carlo algorithm and Unsupervised Latent Concept Modeling methods. Interpreting the plot involves finding the point or range in which the upper plot is minimized and the lower plot is maximized. For the minimization, the minimum happens when the number of topics ≈ 140. Similarly, for the maximization,

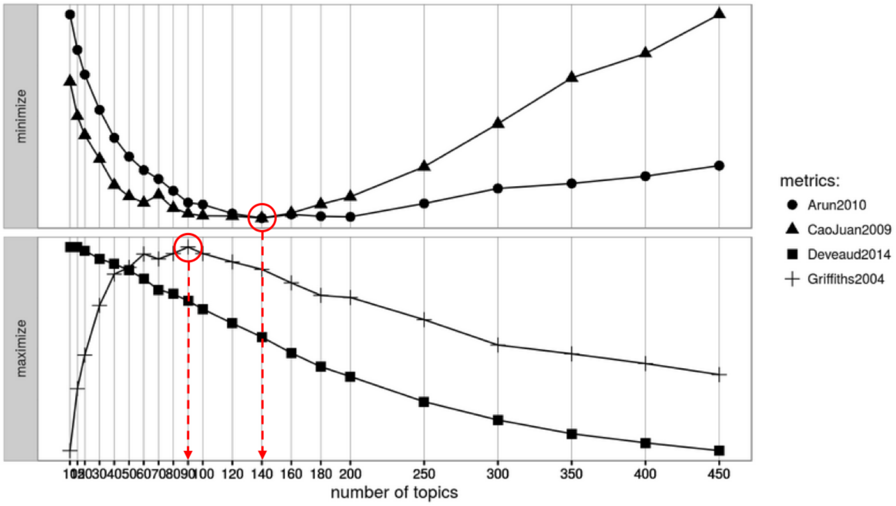


Figure 6. Example LDA Tuning Plot [5]

the maximum happens when the number of topics ≈ 90. Therefore, in this example, the optimal number of topics would range from 90 to 140 [5].

In order to determine the topics associated with a dataset, the optimal number of topics gleaned from the LDA Tuning Plot was fed into the *topicmodels* package [46]. This package was responsible for executing the LDA algorithm, given a number of topics and returning the resulting topic content.

3.5 Sentiment Scoring

To calculate a words sentiment score using a lexicon approach, each individual word from the data-frame must be compared to the words contained within a lexicon dictionary. As mentioned before the data was transformed into a tidy format. In this format, every retained word from the individual tweets is now on an individual line within the dataframe. The data can then be compared to the words found in a lexicon dictionary and return an applicable sentiment score per word. Each of the four dictionaries provides scores in formats:

- The SWN lexicon returns values -1 to 1

- The AFINN lexicon returns values -5 to 5
- The Bing lexicon returns a word as either positive or negative
- The NRC lexicon returns a word as either positive or negative, and also includes the underlying emotion associated with a word.

Once each individual word has been classified, the score is determined per tweet and allows tweets to be compared numerically to other tweets. The score can then be computed as

$$TweetSentimentScore = \sum(words_{pos}) + \sum(words_{neg}), \tag{7}$$

where $words_{pos}$ are the words that are classified with a positive sentiment and $words_{neg}$ are the words that are classified with a negative sentiment. The $TweetSentimentScore$ is the summation of a tweets positive and negative words, which computes the $TweetSentimentScore$ which is either a positive, negative, or zero. For example, the sentence:

> *I really love my dog, he is the best friend anyone could ever ask for!*
>
> When classified with the Bing lexicon, the sentiment would be calculated as:
>
> *x xxxxxx* ***love*** *xx xxx, xx xx xxx* ***best*** *xxxxxx xxxxxx xxxxx xxxx xxx xxx!*

Utilizing Equation 7, the score would be computed as

$$\begin{aligned} &TweetSentimentScore = \sum(love(+1)) + (best(+1)) + \sum(0), \\ &TweetSentimentScore = +2, \end{aligned} \tag{8}$$

where the overall $TweetSentiment$ can be classified

$$TweetSentiment = \begin{cases} positive, if: TweetSentimentScore = + \\ negative, if: TweetSentimentScore = - \\ neutral, else \end{cases} \tag{9}$$

wherein the case of this example, the tweet would be classified as *positive*.

3.6 Visualizations

Calculating data is one aspect of analysis, however insightful and useful visualizations are critical in sharing an understanding of a concept. In this report, knowledge is shared utilizing a number of different visualizations.

- Network Plots
- Bar Charts
- Violin Plots
- Time Series Plots

The network plots were developed in two different areas; Bi-Gram Networks and Correlation Networks. The Bi-Gram plots are based on the number of times a Bi-Gram word combination is used together and displays the relative number of times these word combinations are used together. The Bi-Gram plot allows an analyst to visually observe the interaction between these words and see how some words can form natural clusters. Unlike the Bi-Gram Network, the Correlation Network show how correlated words are together instead of a direct number count. The Correlation Network also helps an analyst visually understand how words are correlated with one another and how words cluster together generally within a certain topic.

The bar charts are relatively simple methods to show the number of times a word has been used within the dataset compared with other words. Additionally, the bar chart is used to display the sentiment distribution across an entire dataset.

The Violin plot is similar to a Box Plot, however, the Violin plot is used because it also displays the relative sentiment distribution across the #hashtags and topics. The Violin plot distribution easily and quickly allows an analyst to understand the range of sentiment values within each #hashtag or topic and understand the general distribution of sentiment scores.

Finally, the Time Series plot clearly allows an analyst to see the change in sentiment over time. This is very insightful because many of the increases and decreases in sentiment values can be shown to occur during key events. Monitoring the changes could allow an analyst to better understand the users' opinion of an event, possibly discover an event, or keep tabs on a particular topic and see how opinion increases or decreases.

3.7 Lexicon Comparison

Previously four different lexicon dictionaries have been described: SWN, AFINN, Bing, and NRC. Originally the SWN lexicon was going to be used in this analysis. However, after a computation time and accuracy test, SWN was determined to not work well in this application.

Computation time was computed by recording the system time at the beginning and the end of the script, the difference between the time was the computation time. The classification

$$Accuracy = \frac{tp + tn}{tp + fp + tn + fn}, \tag{10}$$

was then computed [47]. Where tp = true positive, fp = false positive, tn = true negative, and fn = false negative.

The four test data sets contained tweets or microblog posts that were classified in such a manner as to differentiate between positive, neutral, or negative tweets. During testing,the tp, fp, tn, and fn results could be calculated and compared. The datasets used were:

- The Airline Tweets dataset, contains 14,640 tweets [48].
- The Twitter SA dataset is a 20,000 tweet sample from the original Twitter SA dataset [49].
- The Twitter SA dataset, contains 1,048,576 tweets [49].
- The UM Microblog posts dataset, contains 1,410 tweets [50].

The results of the test can be found in Table 4.

The most significant result in Table 4 is the length of time the SWN lexicon required to calculate the sentiment. The SWN lexicon required each word, parts of speech (POS) to be determined, which drastically increased the computation time. Furthermore, the SWN lexicon generally did not classify Twitter text well. Therefore because of these results, the Bing lexicon was selected due to its overall high classification accuracy and competitive computation time.

Table 4. Lexicon Dictionary Comparison

Dataset Comparison			
Dataset (# tweets)	Lexicon (dictionary)	Time (sec)	Accuracy (%)
Airline Tweets (14,640)	SWN	1141.5720	0.5671
	AFINN	**0.1728**	0.7012
	Bing	1.3367	**0.8239**
	NRC	1.5187	0.5631
Twitter SA Data (20,000)	SWN	1493.7360	0.6150
	AFINN	**0.1343**	0.7146
	Bing	1.2434	**0.7218**
	NRC	1.2996	0.6536
Twitter SA Data (1,048,576)	SWN	N/A**	N/A**
	AFINN	**4.0417**	0.7135
	Bing	74.3100	**0.7251**
	NRC	66.6480	0.6505
UM Microblog Posts (1,410)	SWN	76.9320	0.8458
	AFINN	**0.1022**	**0.9301**
	Bing	0.2328	0.7946
	NRC	0.3896	0.6197

*Bold items denote lowest computation time or highest classification accuracy
**Experiment was not run due to excessive computation time

IV. Analysis

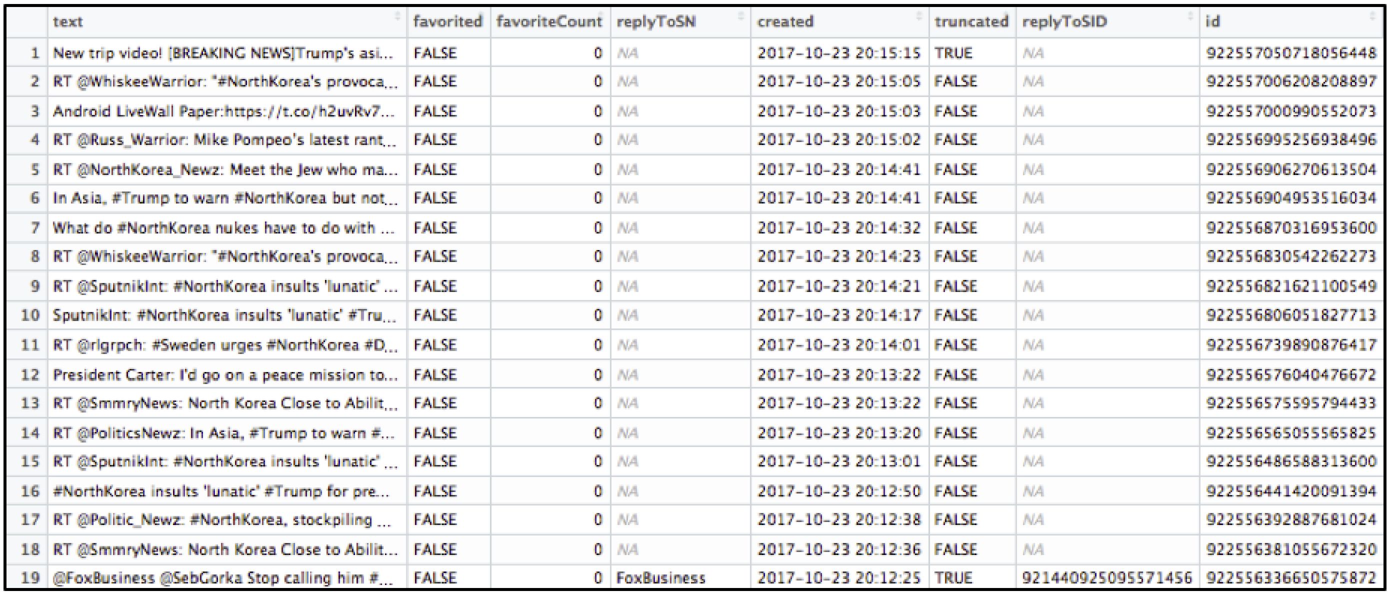
4.1 Analysis Datasets

The datasets acquired to conduct sentiment analysis was collected between 23OCT17 and 07NOV17. The data was collected within a *North Korea* bucket and a *Protest* bucket. The North Korea bucket consists of 11 different #hashtags and the Protest bucket of 10 different #hashtags. The #hastags within each bucket can be found in Table 5.

Table 5. Twitter Data Buckets

North Korea	Protests
#northkorea	#antifa
#nuke	#resistance
#dprk	#facism
#rocketman	#blm
#missile	#blacklivesmatter
#sanctions	#blackpower
#test	#takeaknee
#KimJongUn	#indivisible
#southkorea	#americafirst
#WWIII	#maga
#ww3	

The #hashtags were selected in such a manner as to collect tweets that would have both positive and negative sentiment in order to have a balanced analysis, and also collect tweets that would be grouped within the selected buckets. The tweets collected are a small sample associated with each #hashtag, due to Twitter API limits. Furthermore, during the analysis re-tweeted tweets were maintained in the dataset as long as each re-tweet was re-tweeted by a unique user at a unique time. This was done with the assumption in mind that the re-tweet expressed an opinion that other people felt similar and strongly about, therefore it was included within the dataset. Finally, a sample of the raw data direct from the Twitter API can be observed in Figure 7.



	text	favorited	favoriteCount	replyToSN	created	truncated	replyToSID	id
1	New trip video! [BREAKING NEWS]Trump's asi...	FALSE	0	NA	2017-10-23 20:15:15	TRUE	NA	922557050718056448
2	RT @WhiskeeWarrior: "#NorthKorea's provoca...	FALSE	0	NA	2017-10-23 20:15:05	FALSE	NA	922557006208208897
3	Android LiveWall Paper:https://t.co/h2uvRv7...	FALSE	0	NA	2017-10-23 20:15:03	FALSE	NA	922557000990552073
4	RT @Russ_Warrior: Mike Pompeo's latest rant...	FALSE	0	NA	2017-10-23 20:15:02	FALSE	NA	922556995256938496
5	RT @NorthKorea_Newz: Meet the Jew who ma...	FALSE	0	NA	2017-10-23 20:14:41	FALSE	NA	922556906270613504
6	In Asia, #Trump to warn #NorthKorea but not...	FALSE	0	NA	2017-10-23 20:14:41	FALSE	NA	922556904953516034
7	What do #NorthKorea nukes have to do with ...	FALSE	0	NA	2017-10-23 20:14:32	FALSE	NA	922556870316953600
8	RT @WhiskeeWarrior: "#NorthKorea's provoca...	FALSE	0	NA	2017-10-23 20:14:23	FALSE	NA	922556830542262273
9	RT @SputnikInt: #NorthKorea insults 'lunatic' ...	FALSE	0	NA	2017-10-23 20:14:21	FALSE	NA	922556821621100549
10	SputnikInt: #NorthKorea insults 'lunatic' #Tru...	FALSE	0	NA	2017-10-23 20:14:17	FALSE	NA	922556806051827713
11	RT @rlgrpch: #Sweden urges #NorthKorea #D...	FALSE	0	NA	2017-10-23 20:14:01	FALSE	NA	922556739890876417
12	President Carter: I'd go on a peace mission to...	FALSE	0	NA	2017-10-23 20:13:22	FALSE	NA	922556576040476672
13	RT @SmmryNews: North Korea Close to Abilit...	FALSE	0	NA	2017-10-23 20:13:22	FALSE	NA	922556575595794433
14	RT @PoliticsNewz: In Asia, #Trump to warn #...	FALSE	0	NA	2017-10-23 20:13:20	FALSE	NA	922556565055565825
15	RT @SputnikInt: #NorthKorea insults 'lunatic' ...	FALSE	0	NA	2017-10-23 20:13:01	FALSE	NA	922556486588313600
16	#NorthKorea insults 'lunatic' #Trump for pre...	FALSE	0	NA	2017-10-23 20:12:50	FALSE	NA	922556441420091394
17	RT @Politic_Newz: #NorthKorea, stockpiling ...	FALSE	0	NA	2017-10-23 20:12:38	FALSE	NA	922556392887681024
18	RT @SmmryNews: North Korea Close to Abilit...	FALSE	0	NA	2017-10-23 20:12:36	FALSE	NA	922556381055672320
19	@FoxBusiness @SebGorka Stop calling him #...	FALSE	0	FoxBusiness	2017-10-23 20:12:25	TRUE	921440925095571456	922556336650575872

Figure 7. North Korea Raw Data

4.2 North Korea

Once compiled, the North Korea dataset was found to have the following characteristics displayed in Table 6.

Table 6. North Korea Bucket

Item	Metric
Total Tweets	155,329
Distinct Tweets	72,944
Distinct Words	668,738
Average Words per Tweet	9.17
Scored Words	41,567
% Retained Words	6.22%

The North Korea bucket had a large number of duplicate tweets, as can be seen in the drop from total tweets to the number of distinct tweets within the dataset.

Additionally, the number of scored words within the dataset shows some of the challenges associated with a sentiment analysis lexicon approach, in that a relatively small number of words is used in the end to classify the data.

4.2.1 Data Exploration

Basic data exploration is key to understanding what the data actually contains. In order to accomplish this, the N-grams were explored and any word(s) that required merging were combined. For example, this dataset comprises tweets that contain references to North and South Korea. These terms will be merged and replaced by northkorea and southkorea in order to glean additional information instead of counts of north and south.

4.2.1.1 N-Grams

The dataset being explored was during a tumultuous time surrounding the Korean peninsula. During this time stronger and stronger rhetoric was being exchanged between North Korea and the United States with regards to North Korea's nuclear and ballistic missile tests. In table 7, this can be observed with the word counts of *test, trump, sanctions, kimjongun, missile, nuclear* and *ww3*. Furthermore, the word *namikimdogssk* appears in Uni-Grams and its meaning is unknown by only looking at Uni-Grams.

However, when the Bi-Grams were observed the first entry is not concerning *nuclear war with North Korea*, instead, the *dogmeattrade southkorea* word combination becomes the most common Bi-Gram. Some of the power of this analysis is discovering information that was not necessarily expected. In this case, the *dopgmeattrade* was entirely unexpected. Further observations into the Bi-Grams see a heavy influence of *trump admin*, *admin revokes*, *enactingsanctions trump* and *revokes usborn* word com-

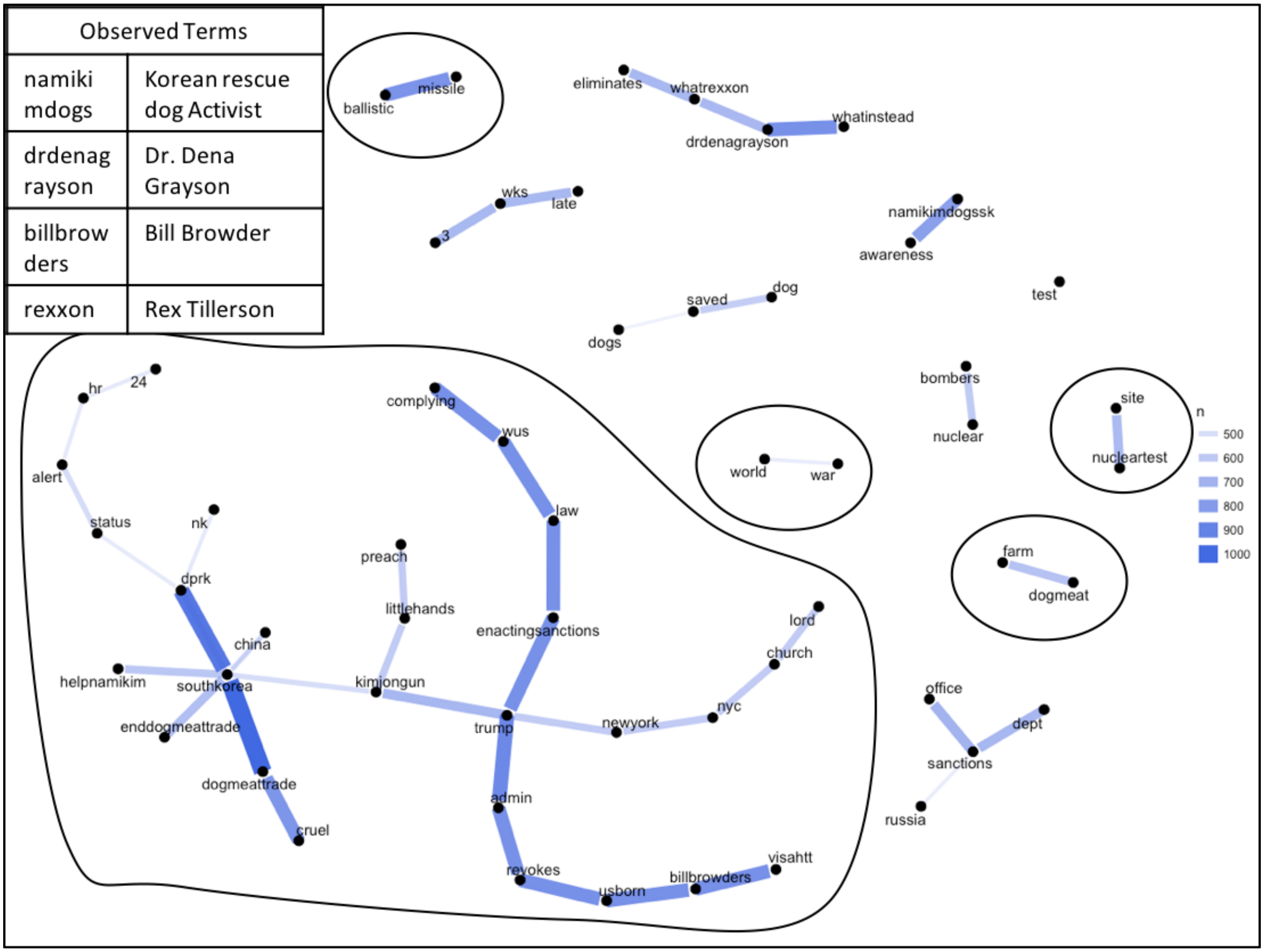
Table 7. North Korea N-Grams

Uni-Gram		Bi-Gram			Tri-Gram			
Word	n	Word 1	Word 2	n	Word 1	Word 2	Word 3	n
southkorea	29888	dogmeattrade	southkorea	1005	admin	revokes	usborn	844
dprk	9380	dprk	southkorea	946	complying	wus	law	844
test	8961	trump	admin	878	enactingsanctions	trump	admin	844
trump	8708	admin	revokes	844	law	enactingsanctions	trump	844
sanctions	7773	billbrowders	visahtt	844	revokes	usborn	billbrowders	844
namikimdogssk	5400	complying	wus	844	trump	admin	revokes	844
kimjongun	5088	drdenagrayson	whatinstead	844	usborn	billbrowders	visahtt	844
missile	4440	enactingsanctions	trump	844	wus	law	enactingsanctions	844
nuclear	3717	law	enactingsanctions	844	dept	sanctions	office	676
ww3	3370	revokes	usborn	844	3	wks	late	674

binations come to the surface, which could allude to a number of decisions President Trump made. Finally when Tri-Grams are investigated more details associated with the Bi-Grams is brought to light. The Tri-Grams, in this case, did not necessarily add new information but instead added more clarification to what was being learned about the data with word combinations of: *admin revokes usborn*, *enactingsanctions trump admin*, and *trump admin revokes.*

4.2.1.2 Network Plots

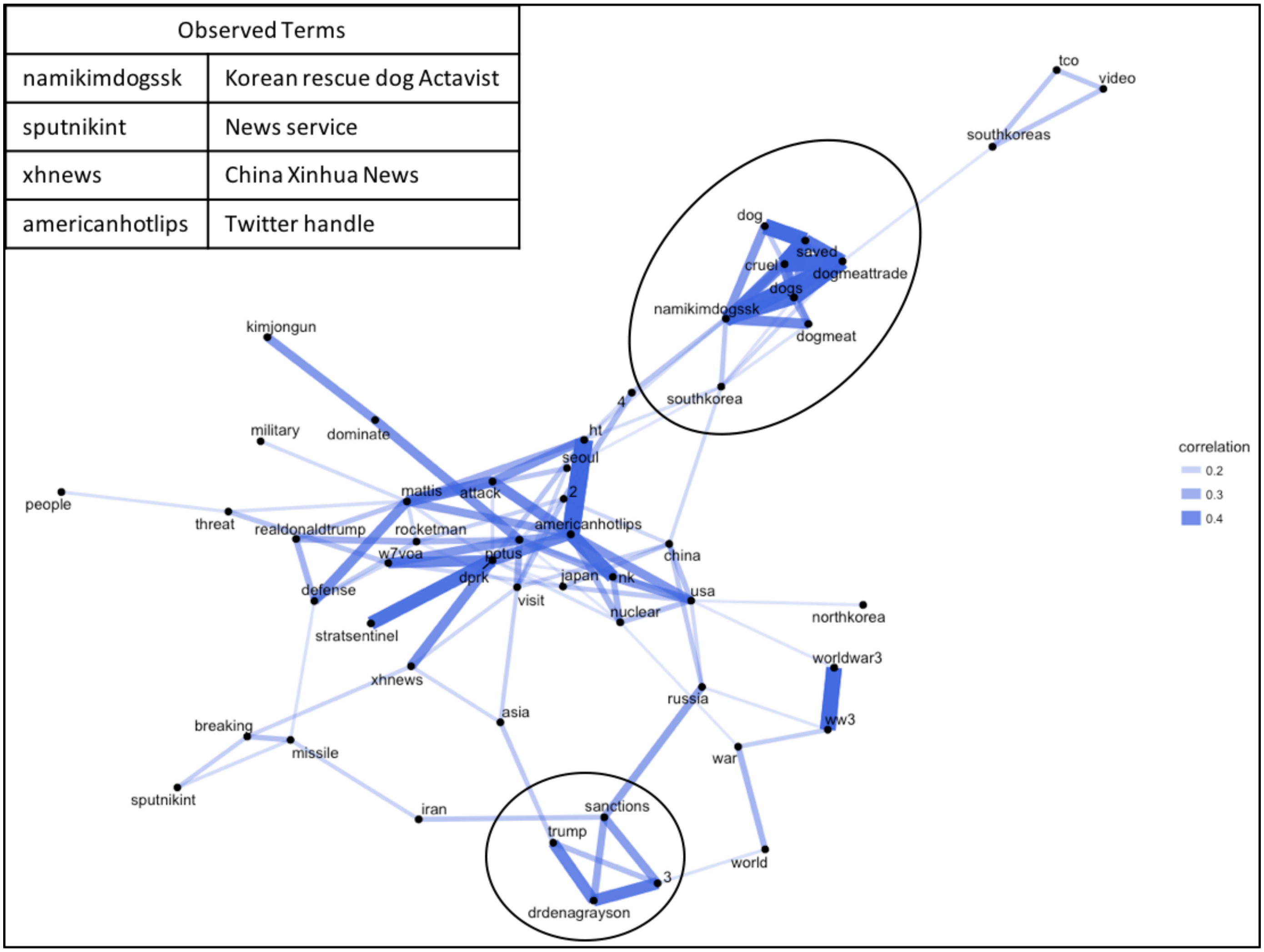
Another manner in which to understand what is going on within the dataset is to observe the Bi-Gram network plots and the correlation plots of the words. In the Bi-Gram network, the interaction between the number of words can be seen in the relative thickness of the linkages between words. In Figure 8, Bi-grams with greater than 400 occurrences are displayed. In this plot, linkages that are thick and dark indicate a large number of occurrences where both words were used together. In the bottom left corner of Figure 8 a large linked network can be seen with topics related to: dprk, kimjonun, trump, enactingsanctions, etc. Furthermore, outside of the large linked cluster, smaller clusters concerning: world war, ballistic missile, farm dogmeat, nucleartest site, etc can also be observed. This is insightful, as it provides information to the analyst about the tweets within the data.



Observed Terms	
namiki mdogs	Korean rescue dog Activist
drdenag rayson	Dr. Dena Grayson
billbrow ders	Bill Browder
rexxon	Rex Tillerson

Figure 8. North Korea Bi-Gram Network

While the Bi-gram Network is helpful in viewing the word interactions, the correlation between words is another metric that will display the connection between multiple words. In Figure 9, the 1,000 most correlated words were retained and those words that were correlated greater than 0.1 were plotted. Similar to how the Bi-Gram network shows the number of linkages between a word with the thickness of the link, the correlation network shows higher correlations with a thicker and darker link. The result is insightful because it displays the mutual relationship between words. Previously in the Uni-Gram exploration, the word *namikimdogssk* surfaced. At the time the word or word phrase did not provide much insight. However, when the word correlation is plotted you can see in the Top rightmost cluster that *namikimdogssk* is highly correlated with the words: *dogmeat, dogmeattrade, dog, cruel, saved, etc.* Now with this further clarity, it is safe to see that within the dataset lies a robust discussion related to the *dogmeatrade* that can be clearly observed in the Correlation Network.



Observed Terms	
namikimdogssk	Korean rescue dog Actavist
sputnikint	News service
xhnews	China Xinhua News
americanhotlips	Twitter handle

Figure 9. North Korea Correlation Network

Additionally, the correlation plot is extremely useful because it furthers the analysts understanding of how different words link together. For example in the bottom center of the network a strong correlation exists between *trump* and *drdenagrayson*. A quick Internet search leads to Dr. Dena Grayson's Twitter account[1]. A cursory search through her Twitter account reveals that Dr. Dena Grayson is a very outspoken critic of President Trump and furthermore within the acquired Twitter data all of her comments are either directed towards President Trump or Donald Trump Jr., which explains the high correlation between both *trump* and *drdenagrayson*.

4.2.2 #Hashtag Sentiment Analysis

In order to calculate the sentiment of the data set, the scores must be calculated. In order to accomplish this, the Bing lexicon was used. As seen in Table 6, one of the challenges with a lexicographical approach is the reliance on the lexicon dictionary. In the North Korea Bucket, only 6.22% of words were used to classify the tweets.

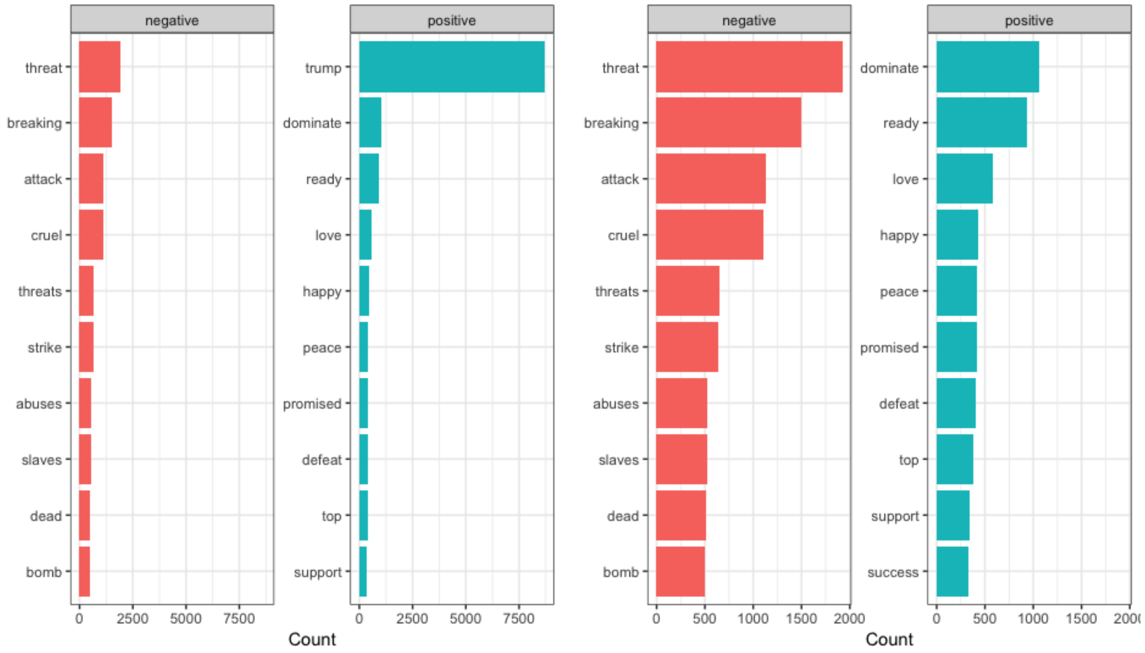
Once the scores are computed, dataset sentiment can be investigated.

In Figure 10a, the top ten most positive and negative words can be seen. However, the top positive word in the chart is referring to *President Trump* instead of the word *trump* and is skewing the results of the chart. Therefore in Figure 10b, *trump* has been removed to show a more accurate representation of the positive and negative words.

In order to understand if the dataset is generally positive or negative, the distribution of TweetSentiments can be observed. In Figure 11 the distribution of TweetSentiments is generally negative. Therefore we can conclude that overall the sentiment of this dataset is negative. However it can also be seen that the distribution has a decidedly bi-modal distribution with a large quantity of positive tweets within the dataset, so it cannot be inferred that all tweets with regards to this data are negative.

Furthermore, the individual #hashtag distributions can be found in the Figure 12. In the Violin Plot, the distribution of scores per #hashtag are clearly seen and the

[1] `https://Twitter.com/drdenagrayson`



(a) With “trump”

(b) Without “trump”

Figure 10. North Korea Most Popular Positive and Negative Words

chevron symbol represents the median of the data which helps very quickly identify whether a #hashtag is considered positive or negative. From Figure 12 the content within the #hashtags can be considered either positive, negative or neutral and the overall sentiment classification of each #hashtag can be found in Table 8.

Table 8. North Korea #Hashtag Classification

Negative	Neutral	Positive
WWIII	rocketman	test
ww3		sanctions
southkorea		KimJongUn
nuke		
northkorea		
missile		
dprk		

From Table 8, it is interesting to see that *KimJongUn* is classified as positive. The reason for this may be that the lexicon has a difficult time classifying sarcasm

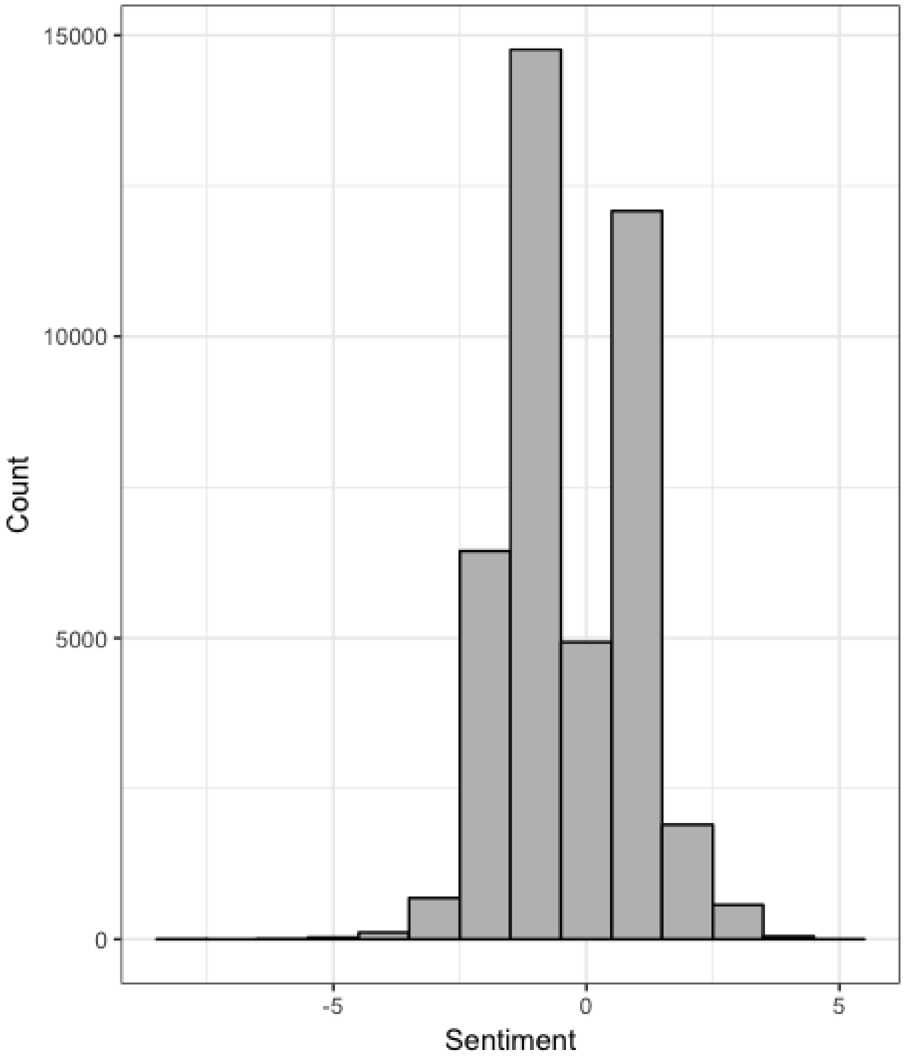


Figure 11. North Korea $TweetSentimentScore$ **Distribution**

that is present within the tweets.

In Table 9 the most negative and positive tweets can be observed. Generally speaking, the negative tweets have some very colorful negative language and the positive have some element of sarcasm present.

The change in sentiment over time is a particularly insightful visualization of the $TweetSentiment$ and is visualized in Figure 13. The $TweetSentimentScore$ was grouped by each #hashtag and for each day the $TweetSentimentScore$ was summed together for a daily score. Each daily score was then plotted to show the variation of

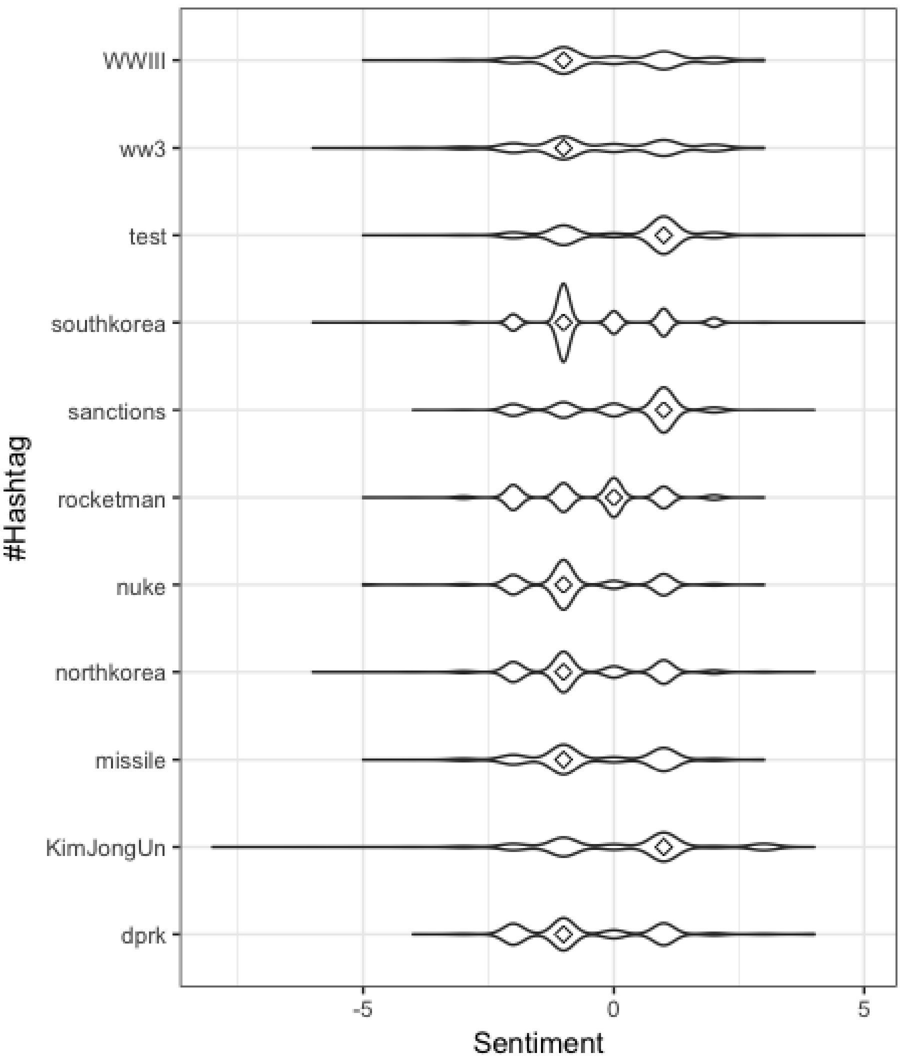


Figure 12. North Korea Violin Plot

the $TweetSentimentScore$ over time.

The results of this analysis are very insightful because it shows the change in sentiment over time and shows how people voiced their opinions about the world around them. For example:

1. 27OCT17: State Department gives Congress list of Russia sanction targets[2]. Which corresponds to an increase in positive sentiments in #sanctions and # WWIII.

[2] `https://www.cbsnews.com/news/state-dept-gives-congress-list-of-russia-sanction-targets/`

Table 9. North Korea Positive and Negative Tweets

Negative Tweets
1. #vnk #KimJongUn #trash #pig #hiding #dumpster #hill belly #fat #gook #moron #idiot #trash #die #soon #loser #with #your #people
2. Fat f**ker on a pedestal is ruining the world with his gay boy bait fat, lump of lard s**t gay loving c**t f**k sack s**t. #KimJongUn
3. Gay boy Kim Jong gay boy like a Eminem suck a d**k boy. F**k you gay fat prick a fat s**t nazi s**t. #KimJongUn
Positive Tweets
1. @realDonaldTrump Wow! So stunning it gave me chills! What a beautiful amazing welcome. Truly memorable... #south_korea #historical #usa
2. @DrDenaGrayson #Trump owes a lot of $ 2 #Russian Oligarchs #Putin wanted #Sanctions gone & helped #Trump win Presid...
3. #ZippytheP 171102TH Boom goes Pyongyang, boom Seoul, boom LA & boom #TrumpTower? #realDonaldTrump #POTUS...

2. 31OCT17: There was a large tunnel collapse at the site of North Koreas nuclear testing facility in which there were an estimated 200 workers killed in the cave-in[3]. Which corresponds to an increase in negative sentiments for #dprk, #nuke, #northkorea, #rocketman, and #ww3 hashtags.

3. 04NOV17: Saudi Arabia intercepted a ballistic missile over its capital[4]. Which corresponds to a sharp increase in negative sentiment for the #missile hashtag.

Figure 13 is very insightful because with a few lines of code a casual observer, Government Agency or Polling institute can very easily visualize the sentiment of an event and can easily relate that increase or decrease in positive or negative sentiment with a real-world event in fractions of the time it would take to acquire otherwise. However, the data was acquired specifically by collecting data related to specific

[3] `http://www.foxnews.com/world/2017/10/31/200-feared-dead-after-tunnel-collapses-at-north-korean-nuclear-test-site-japanese-tv-claims.html`

[4] `https://www.nytimes.com/2017/11/04/world/middleeast/missile-saudi-arabia-riyadh.html`

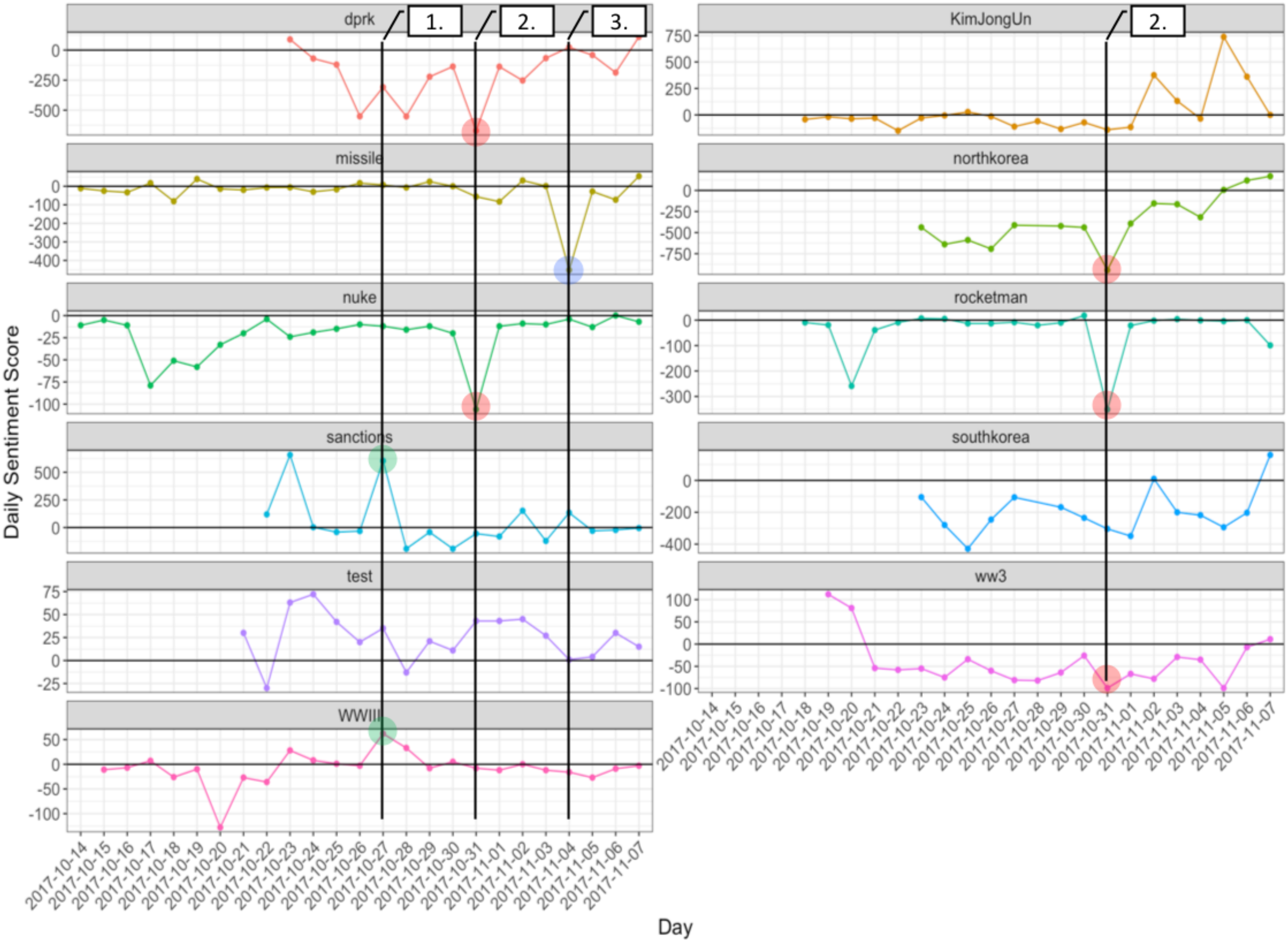


Figure 13. North Korea Hashtag Time Series

#hashtags. If data was not able to be acquired in this way, topic modeling could be used to determine the underlying topics and then rerun the analysis.

4.2.3 Topic Analysis Sentiment Analysis

Previously the analysis was conducted utilizing #hashtags to differentiate between groups of tweets. However, this was more a function of how the Twitter data could be easily acquired to ensure they would generally fit within the North Korea bucket. A more realistic method to conduct sentiment analysis is one in which a large Twitter dataset is acquired without regard to #hashtags or possibly a sampling of tweets acquired directly through Twitter or a third party.

The sentiment could be calculated based off all the tweets within this new dataset, but it would not provide as much detail because the underlying topics of the tweets would be unknown.

In order to discover the hidden topics within a corpus of text, the LDA method will be used. However, the challenge with utilizing LDA is selecting the number of topics. In the North Korea dataset, the data was acquired from 11 different #hashtags. Therefore 11 topics could be selected, however, this may return similar information that was previously discovered. In an effort to discover hidden topics that may not be observed directly within each #hashtags an LDA Tuning algorithm was used to determine an optimal number of hidden topics within the dataset. The algorithm is time-consuming, but it can be run in parallel on a multi-core processor to decrease computation time. As a benchmark, using a 2017 MacBook Pro with an Intel i5 3.1 GHz dual-core processor and 8GB of Memory, the LDA tuning algorithm took approximately 15-20 minutes to be computed.

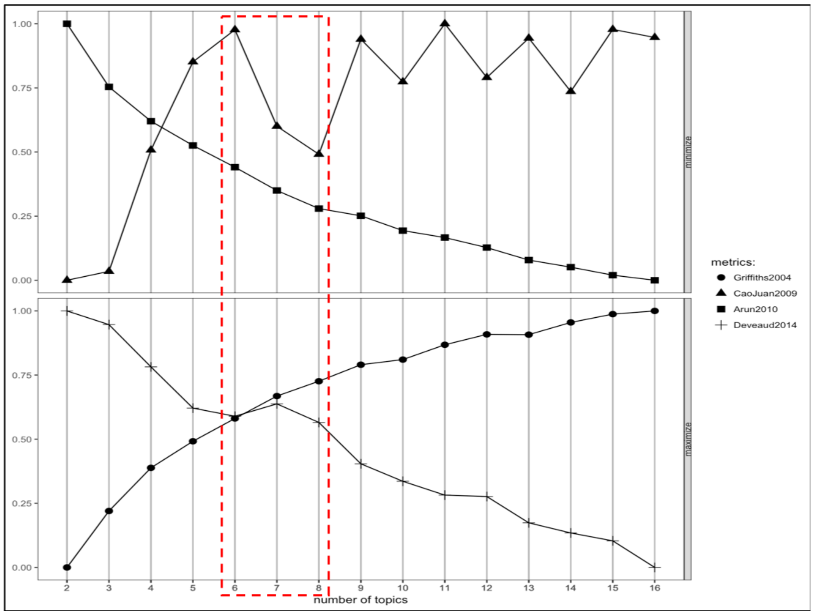


Figure 14. North Korea LDA Tuning Plot

The results from the LDA tunning graph can be found in Figure 14. The plot shows that both minimization and maximization cross with a recommended number of clusters ≈ 6. Because of the divergence between the methods, the art of selecting the number of topics comes into play. The LDA algorithm was run with clusters = 6, 7 and 8 then manually compared to each other.

If for instance, a subject matter expert for the Korean peninsula was involved, that individual could assist an analyst make a more informed decision on the optimal number of topics to select by comparing the results of selecting different amounts of clusters. For the purpose of the North Korea Bucket, clusters = 7 was chosen, as it appeared to have a balance between the number of topics and overlap between the topics, and can be seen in Table 10. With 7 topics selected, the topics were renamed to assist the analyst in understanding what the topics include and can be seen in Table 10.

Instead of #hastags, the Topics sentiment distributions can be found in the Figure 15. From Figure 15 the content within the Topics can be considered either positive, negative or neutral and the overall sentiment classification of each Topic can be found in Table 11.

From Table 11, it is interesting to see that only a single topic is classified as

Table 10. North Korea Bucket Modeled with 7 Topics

Number	test	missile	pdjt	southkorea	dogmeat	sanctions	nuke
1	test	missile	southkorea	dprk	southkorea	trump	ww3
2	wkt	kimjongun	realdonaldtrump	southkorea	namikimdogssk	sanctions	nuclear
3	6s	rocketman	visit	japan	dogmeattrade	amp	war
4	follow	breaking	korea	potus	dogs	iran	china
5	live	2	seoul	threat	dog	russia	usa
6	xhnews	sputnikint	asia	mattis	tco	htt	world
7	house	kim	president	stratsentinel	video	drdenagrayson	nuke
8	odi	potus	report	ht	southkoreas	people	northkorea
9	youre	ballistic	whitehouse	attack	cruel	nucleartest	wwiii
10	cricket	riyadh	maga	defense	dogmeat	site	nk

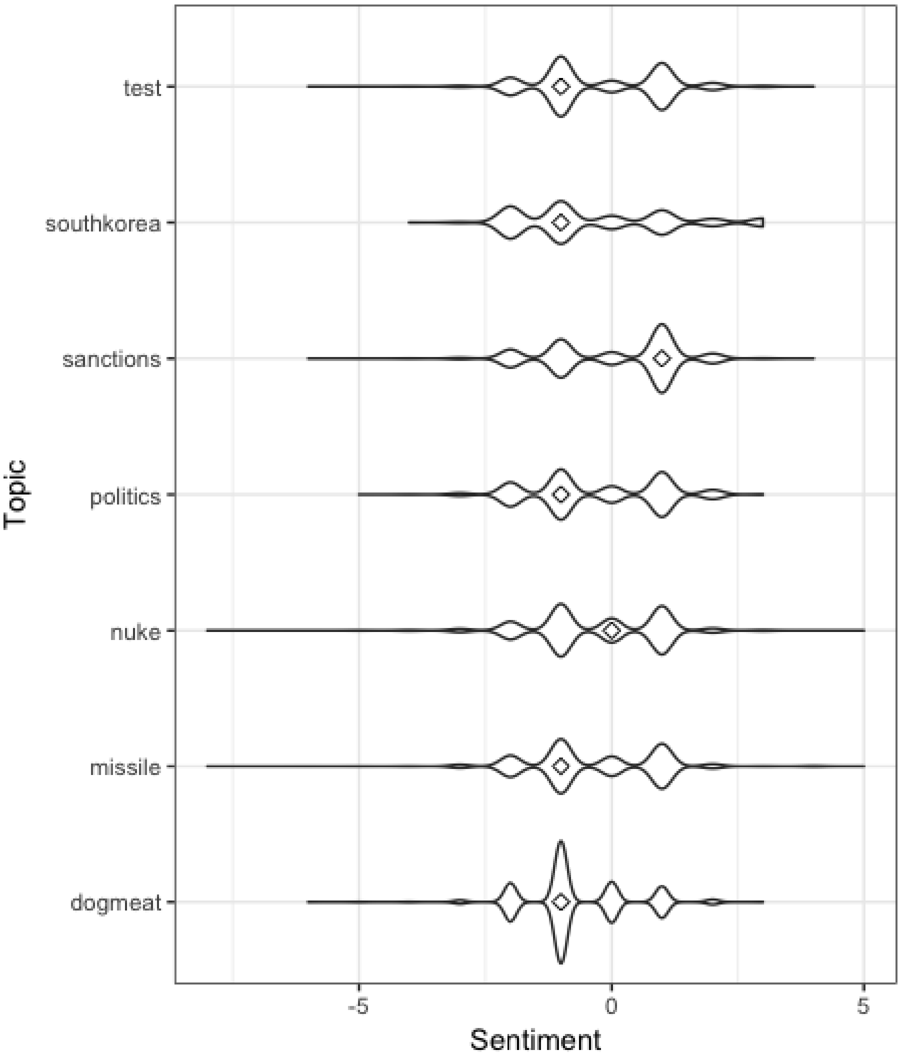


Figure 15. North Korea Topic Violin Plot

Table 11. North Korea #Hashtag Topic Classification

Negative	Neutral	Positive
test	nuke	sanctions
southkorea		southkorea
politics		politics
missile		missile
dogmeat		dogmeat

positive, and all remaining topics but one are negative. It can be inferred that because *sanctions* has a positive sentiment classification, the topic had a positive opinion in regards to sanctions.

As previously mentioned the change in sentiment over time is a particularly insightful visualization of the *TweetSentiment* and is visualized in Figure 16. In this case, the *TweetSentimentScore* was grouped by each Topic previously discovered and for each day the *TweetSentimentScore* was summed together for a daily score.

The results of this analysis aesthetically different from the time series seen in Figure 13 however very similar information can be gleaned as seen in Figure 16. For example:

1. 23OCT17: America prepares to put nuclear-armed bombers on 24-hour alert for the first time since 1991[5]. Also, a discussion concerning President Trump's actions pertaining to sanctions is also discussed. These corresponds to large positive sentiments in the *trump* and *nuclear* topics.

2. 27OCT17: Secretary of Defense Mattis visits the DMZ[6]. Which corresponds to a large negative sentiment in the *politics* topic.

3. 31OCT17: There was a large tunnel collapse at the site of North Koreas nuclear testing facility in which there were an estimated 200 workers killed in the cave-in[7]. Which corresponds to an increase in negative sentiments in the *missile* and *politics* topics.

4. 01NOV17: Large number of tweets condemning the dogmeat trade, corresponding to a large negative sentiment in the *dogmeat* topic.

[5] http://www.telegraph.co.uk/news/2017/10/23/america-prepares-put-nuclear-armed-bombers-24-hour-alert-first/

[6] http://www.cnn.com/2017/10/26/politics/mattis-south-korea-dmz/index.html

[7] http://www.foxnews.com/world/2017/10/31/200-feared-dead-after-tunnel-collapses-at-north-korean-nuclear-test-site-japanese-tv-claims.html

5. 04NOV17: Saudi Arabia intercepted a ballistic missile over its capital[8]. Which corresponds to a sharp increase in negative sentiment for the #missile hashtag.

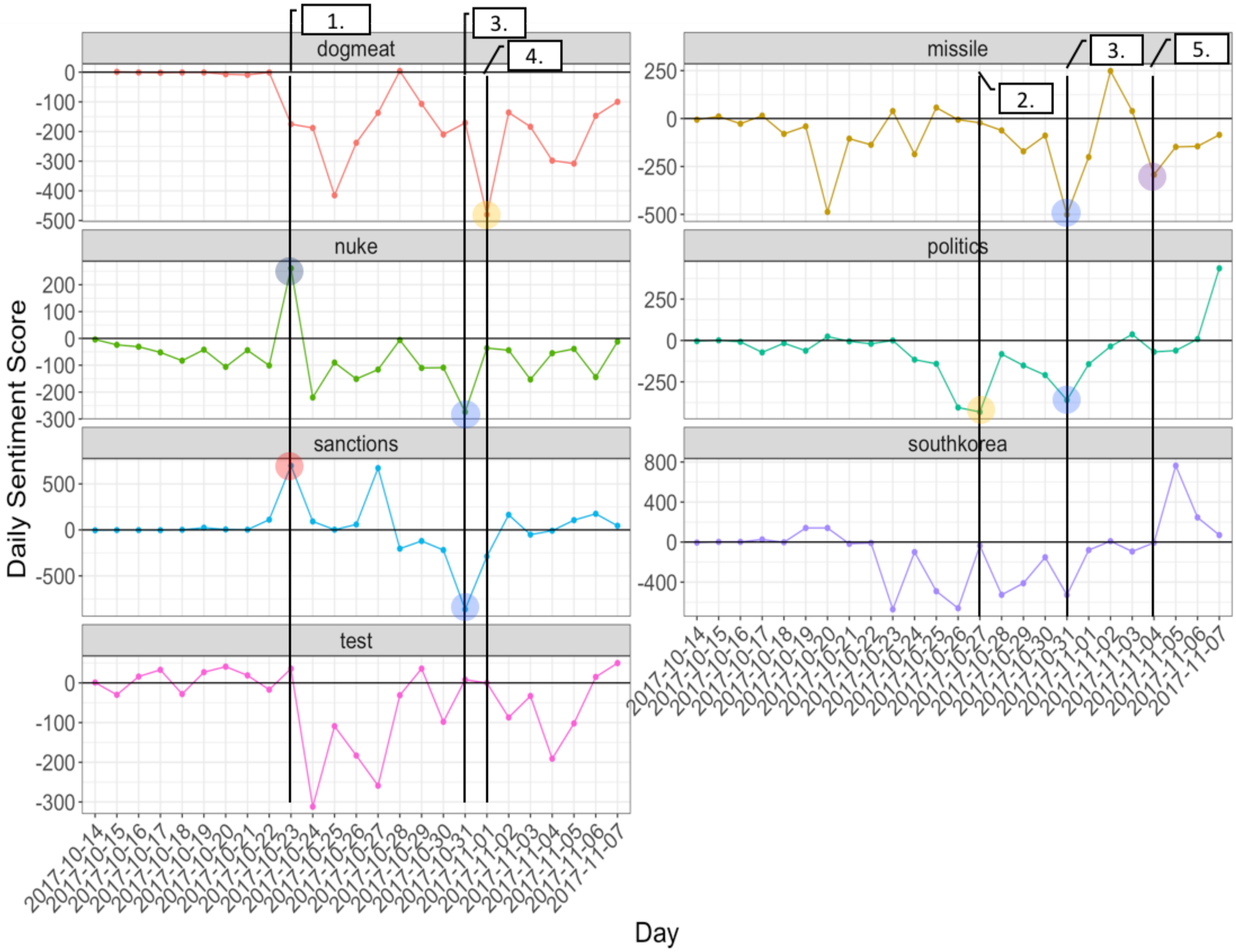


Figure 16. North Korea Topic Time Series

Sentiment analysis in concert with LDA is an incredibly powerful tool. Utilizing LDA, eight topics were selected to represent the North Korea Bucket as opposed to the original 11 #hashtags and similar results were reproduced with both methods.

[8] https://www.nytimes.com/2017/11/04/world/middleeast/missile-saudi-arabia-riyadh.html

4.3 Protest

Once compiled, the Protest dataset was found to have the following characteristics displayed in table 12.

Table 12. Protest Bucket

Item	Metric
Total Tweets	150,000
Distinct Tweets	124,199
Distinct Words	1,124,894
Average Words per Tweet	9.06
Scored Words	84,469
% Retained Words	7.51%

The Protest bucket had a much smaller number of duplicate tweets and only lost $\approx 26,0000$ tweets, as opposed to the North Korea bucket that lost closer to $\approx 82,000$ tweets. Additionally, the Protest dataset had nearly double the amount of scored words as compared to the North Korea bucket.

4.3.1 Data Exploration

As previously, N-grams will be explored, however for the Protest bucket, there were no terms that appeared to require merging.

4.3.1.1 N-Grams

The Protest dataset was harvested when a number of different groups were vocal in their opinions. During the time frame, Black Lives Matters, Antifa (Anti-fascism), those for and against President Trump and NFL players taking a knee all occurred. In Table 13, *maga* translates to "Make America Great Again", which was President Trump's campaign slogan during the 2016 Presidential race. Furthermore *blm* translates to "black lives matter".

Table 13. Protest N-Grams

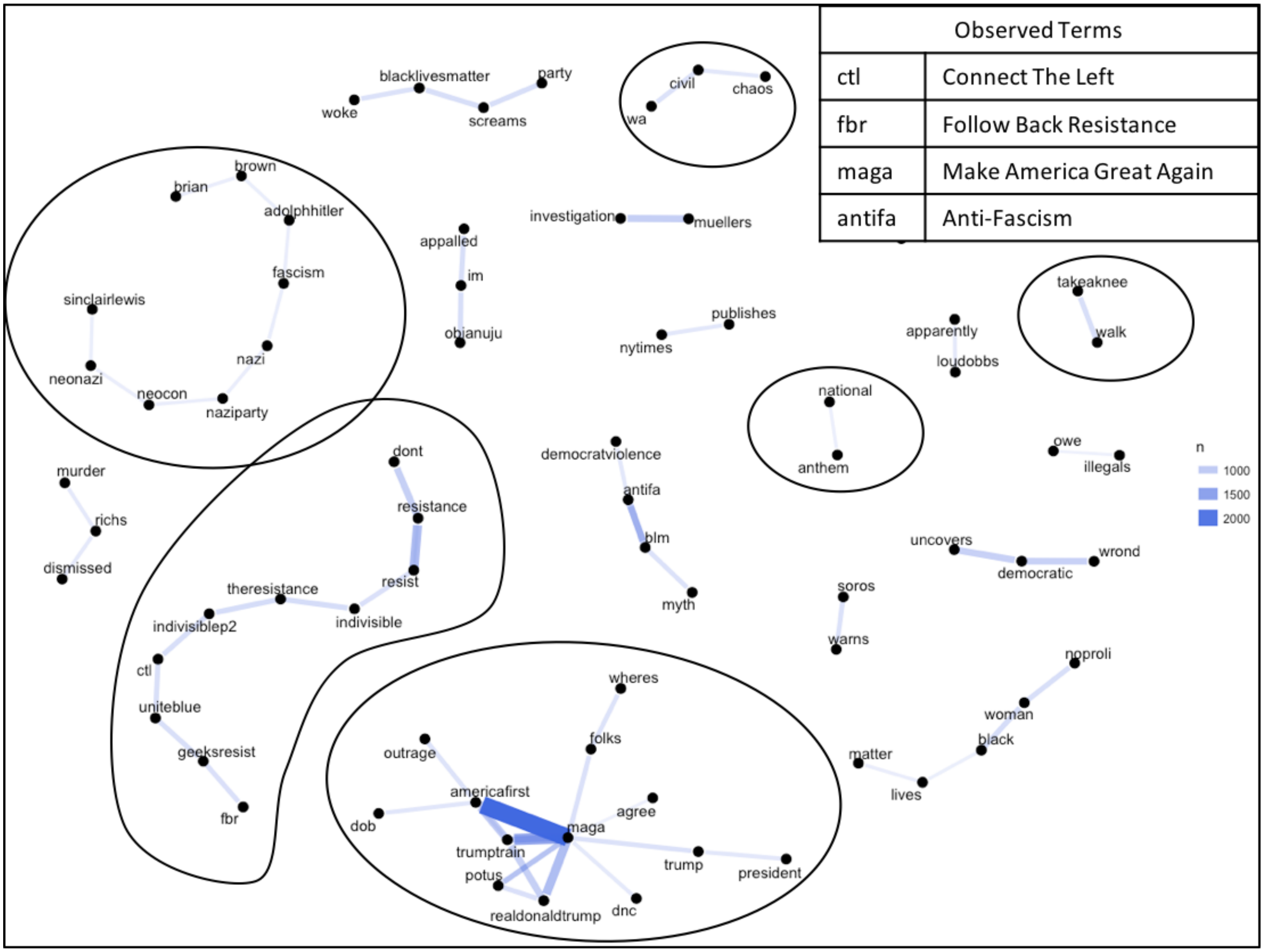
Word	n	Word 1	Word 2	n	Word 1	Word 2	Word 3	n
maga	21192	maga	americafirst	2151	uncovers	democratic	wrond	921
antifa	17187	maga	trumptrain	1431	maga	realdonaldtrump	trumptrain	905
resistance	13924	resist	resistance	1196	realdonaldtrump	trumptrain	americafirst	812
takeaknee	13600	maga	realdonaldtrump	1164	theresistance	indivisiblep2	ctl	750
blacklivesmatter	13539	page	ad	1084	party	screams	blacklivesmatter	749
blm	12780	trumptrain	americafirst	992	black	woman	noproli	747
americafirst	11956	muellers	investigation	941	obianuju	im	appalled	747
trump	10546	realdonaldtrump	trumptrain	940	ctl	uniteblue	geeksresist	723
realdonaldtrump	8639	antifa	blm	936	uniteblue	geeksresist	fbr	723
indivisible	8333	resistance	dont	934	indivisiblep2	ctl	uniteblue	722

The Bi-Grams were interesting because five of the top ten entries were combinations of President Trump supporters phrases which include: *maga, americafirst, trumptrain*, and *realdonaldtrump.* The remaining Bi-Grams are more vague references to: *resist, resistance, antifa*, and *blm.* The Tri-Grams are very interesting in the Protest bucket. Some additional combinations of *maga, realdonaldtrump, trumptrain*, and *americafirst* appear. Interestingly a couple of references to *ctl*, and *fbr* appear. Originally it was thought that these were incorrectly cleaned words, but upon further investigation, *ctl* is #ctl, which is understood to mean *ConnectTheLeft*, which is a #hashtag used to "bringing lefties, progressives, liberals, and Democrats together to fight the insanity and Un-American actions of the Republican Tea Party[9]" which is also in conjunction with #uniteblue. Additionally *fbr* corresponds #fbr, which is *FollowBackResistance* and is related with resistances activities across social media. Finally, the Tri-Grams pulled in *obianuju* which is the Twitter handle of @obianuju, who posted a tweet explaining how appalled she was in the way that an elected State Representative acted towards one of his constituents. In the Protest bucket, her tweet was re-tweeted $\approx$ 750 times and appears to show many people agreed with her comment.

[9] `https://tagdef.com/en/tag/ctl`

4.3.1.2 Network Plots

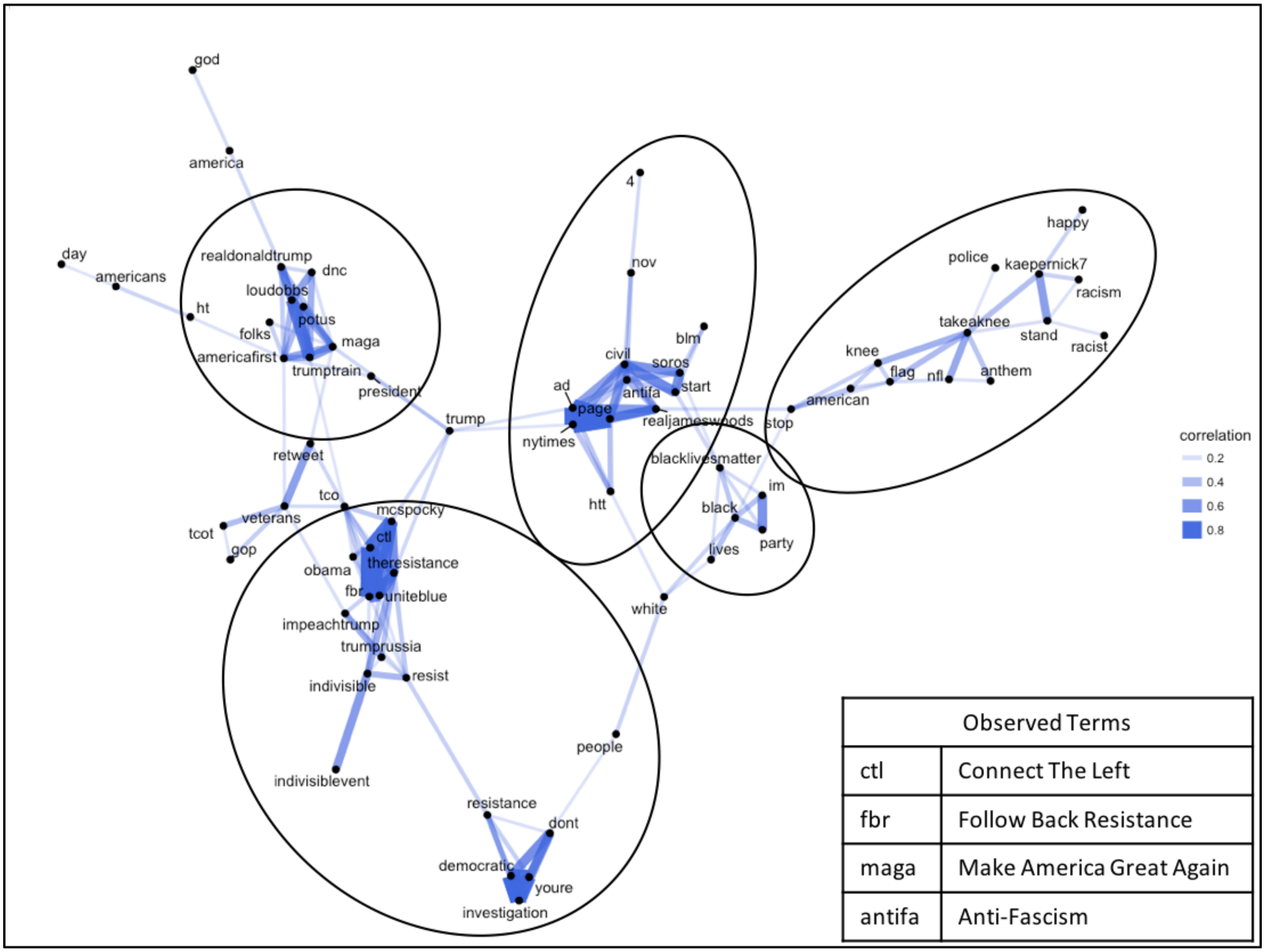
In the bottom center corner of Figure 17 a linked network can be seen with topics related to: americafirst, maga, trumptrain, etc. Another cluster pertaining to: theresistance, ctl, fbr, etc can also be found. Additionally, a cluster in the top left corner of the chart has a number of words pertaining to: adolphhitler, fascism, nazi, neonazi, etc. Furthermore, outside of the large linked cluster, smaller clusters concerning: civil chaos, takeaknee walk, national anthem, etc can also be observed. Which shows a wide variety of different protests occurring within the dataset.



Observed Terms	
ctl	Connect The Left
fbr	Follow Back Resistance
maga	Make America Great Again
antifa	Anti-Fascism

Figure 17. Protest Bi-Gram Network

In Figure 18, the 1100 most correlated words were retained and those words that were correlated greater than 0.1 were plotted. This correlation diagram is interesting because most of the different protests are clustered together and are generally separate from one another. The data generally cluster into five different clusters which could generally be summarized by: trump, resistance, blacklivesmatter, antifa, and nfl. Which is what we have identified in earlier results.

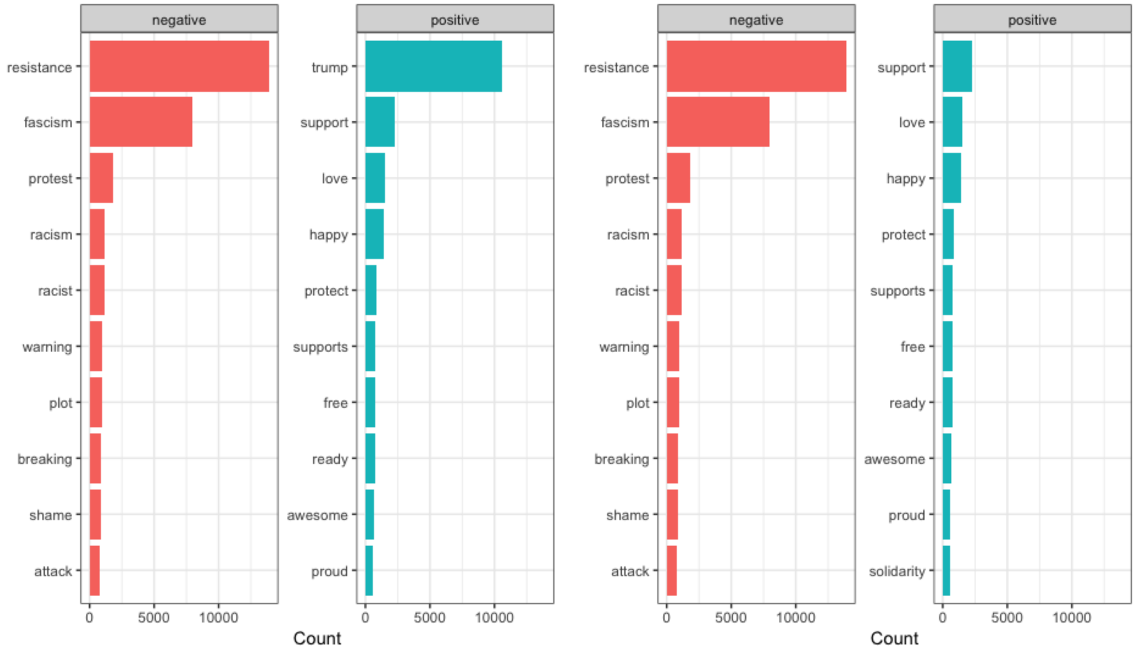


Observed Terms	
ctl	Connect The Left
fbr	Follow Back Resistance
maga	Make America Great Again
antifa	Anti-Fascism

Figure 18. Protest Correlation Network

4.3.2 #Hashtag Sentiment Analysis

Once the scores are computed, the dataset sentiment can be investigated.



(a) With "trump"

(b) Without "trump"

Figure 19. Protest Most Popular Positive and Negative Words

Like in the North Korea Dataset *trump* was a highly used word. In Figure 19a, the top ten most positive and negative words can be seen. However, the top positive word in the chart is referring to *President Trump* instead of the word *trump* and is skewing the results of the chart. Therefore in Figure 19b, *trump* has been removed to show a more accurate representation of the positive and negative words.

In Figure 20 the distribution of TweetSentiments is generally negative. Therefore we can conclude that overall the sentiment of this dataset is negative. However it can also be seen that the distribution has a bi-modal distribution with a portion of positive tweets within the dataset, so it cannot be inferred that all tweets with regards to this data are negative.

Furthermore, the individual #hashtag distributions can be seen in the Figure 21.

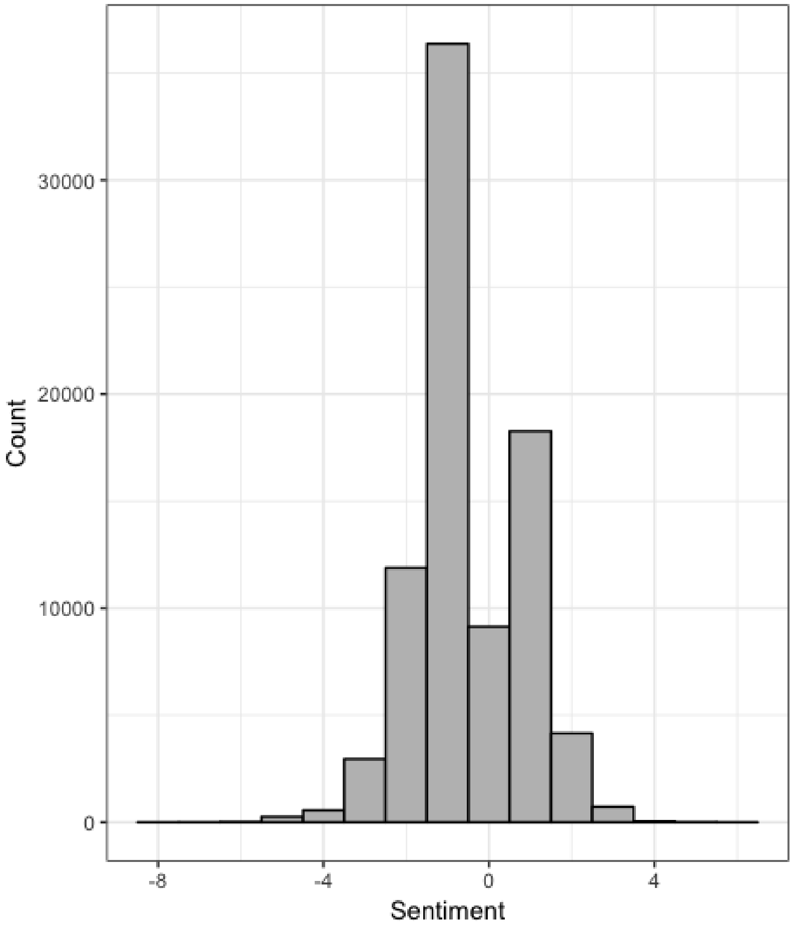


Figure 20. Protest $TweetSentimentScore$ **Distribution**

From Figure 21 the content within the #hashtags can be considered either positive, negative or neutral and the overall sentiment classification of each #hashtag can be found in Table 14.

Table 14. Protest #Hashtag Classification

Negative	Neutral	Positive
takeaknee	maga	indivisible
resistance	blackpower	
fascism	americafirst	
blm		
blacklivesmatter		
antifa		

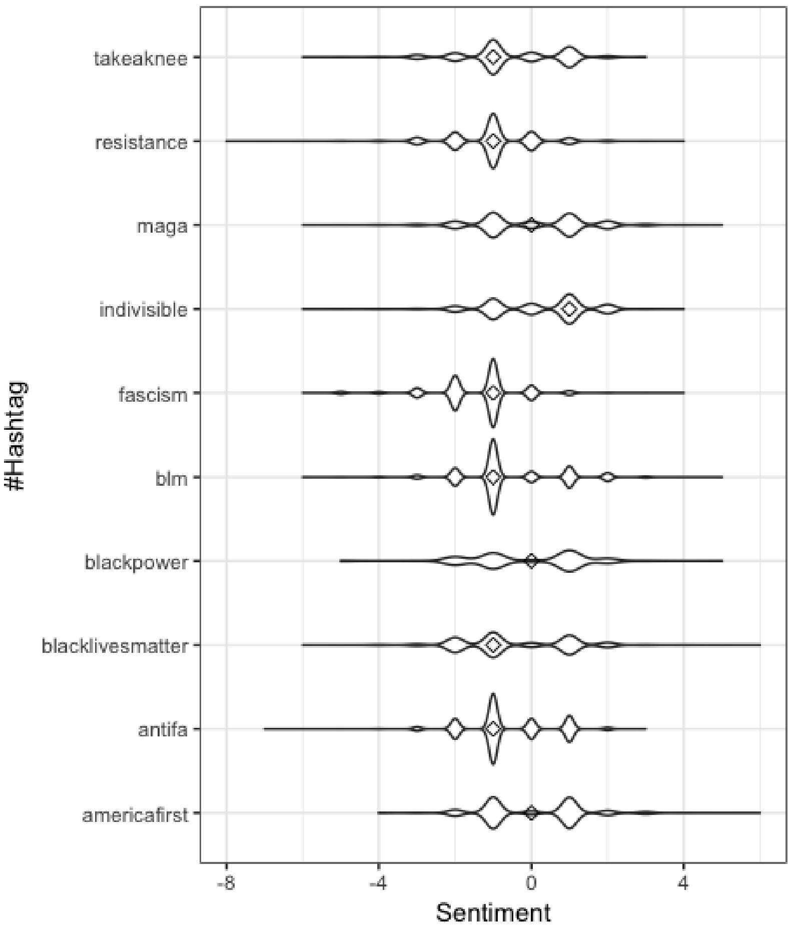


Figure 21. Protest Violin Plot

From Table 14, it is interesting and possibly not surprising to see that the majority of the #hashtags are classified as neutral or negative, with *indivisible* being the only positive #hashtag.

In Table 15 the most negative and positive tweets can be observed. Generally speaking these tweets are much less vulgar than those found in the North Korea bucket, however, they are still full of negative and positive sentiment words.

The change in sentiment over time is a particularly insightful visualization of the $TweetSentiment$ and is visualized in Figure 22. The $TweetSentimentScore$ was grouped by each #hashtag and for each day the $TweetSentimentScore$ was summed together for a daily score. Each daily score was then plotted to show the variation of

Table 15. Protest Positive and Negative Tweets

Negative Tweets
1. Reckless, outrageous; undignified behavior is reckless, outrageous undignified; dangerous to America-Sen. Flake 2. Don't confuse bat shit crazy with Conservative! Roy Moore is bat shit crazy! Jeff Flake is Conservative! #Resistance #TrumpisaMoron 3. Anti hate activist while spewing hate. Resisting hate while creating hate. Sounds like a vicious circle of hate to me! #Resistance
Positive Tweets
1. An inspiring and profound evening. Congratulations to #BlackLivesMatter on winning this year's Sydney Peace Prize. Keep up the amazing work! 2. I love #makeup I love #poltics I love #goodbooks I love #history I love #feminsm #BLM #LGBTQ Women are complex and thats beautiful 3. RT @BrownGirlBegins: #BlackLivesMatter awarded Sydney Peace Prize. Congratulations @opalayo @OsopePatrisse @aliciagar inspiring. @blavity

the $TweetSentimentScore$ over time.

The results of this analysis over time can be found in Figure 22 and shows how opinions changed through time. Notable events were:

1. 29OCT17: NFL Protests Reach A Boiling Point, As Players, Owners Cancel Social Concerns Meeting[10]. Which corresponds to spikes in #blacklivesmatter, #blm, and #takeaknee.

2. 29OCT17: Spanish Court Nullifies Catalonia Independence As Civil War Looms[11]. Which adds to the spike in #fascism.

3. 29OCT17: An individual boarding her Amtrak train was told to remove her "love trumps hate" political button because Amtrak federally funded which caused a vocal outcry on Twitter[12]. Which adds to the spike in #fascism.

4. 03NOV2017: Antifa Civil War on November 4 Was Really Just a Few Protests

[10] http://deadline.com/2017/10/nfl-anthem-protests-reach-boiling-point-1202197034/

[11] http://yournewswire.com/spain-nullifies-catalonia-independence-civil-war/

[12] https://Twitter.com/feytwee/status/923990107656507392

Against Trump[13] which was a planned set of marches and demonstrations corresponding to #antifa and #resistance.

5. 05NOV17: Reactions to the lackluster protests and demonstrations that had been planned for Antifa's Civil War on November 4. Which corresponds to #antifa, #resistance, and #maga.

The Protest dataset was very different as compared to the North Korea dataset. Fewer news articles directly corresponded to events and more events were only observed on Twitter. For example, the Amtrak event was started with the Twitter post seen in Figure 23. This event did not make it to mainstream news sources but was re-tweeted over 4,500 times.

[13] http://www.newsweek.com/antifa-civil-war-november-4-really-just-few-protests-against-trump-702150

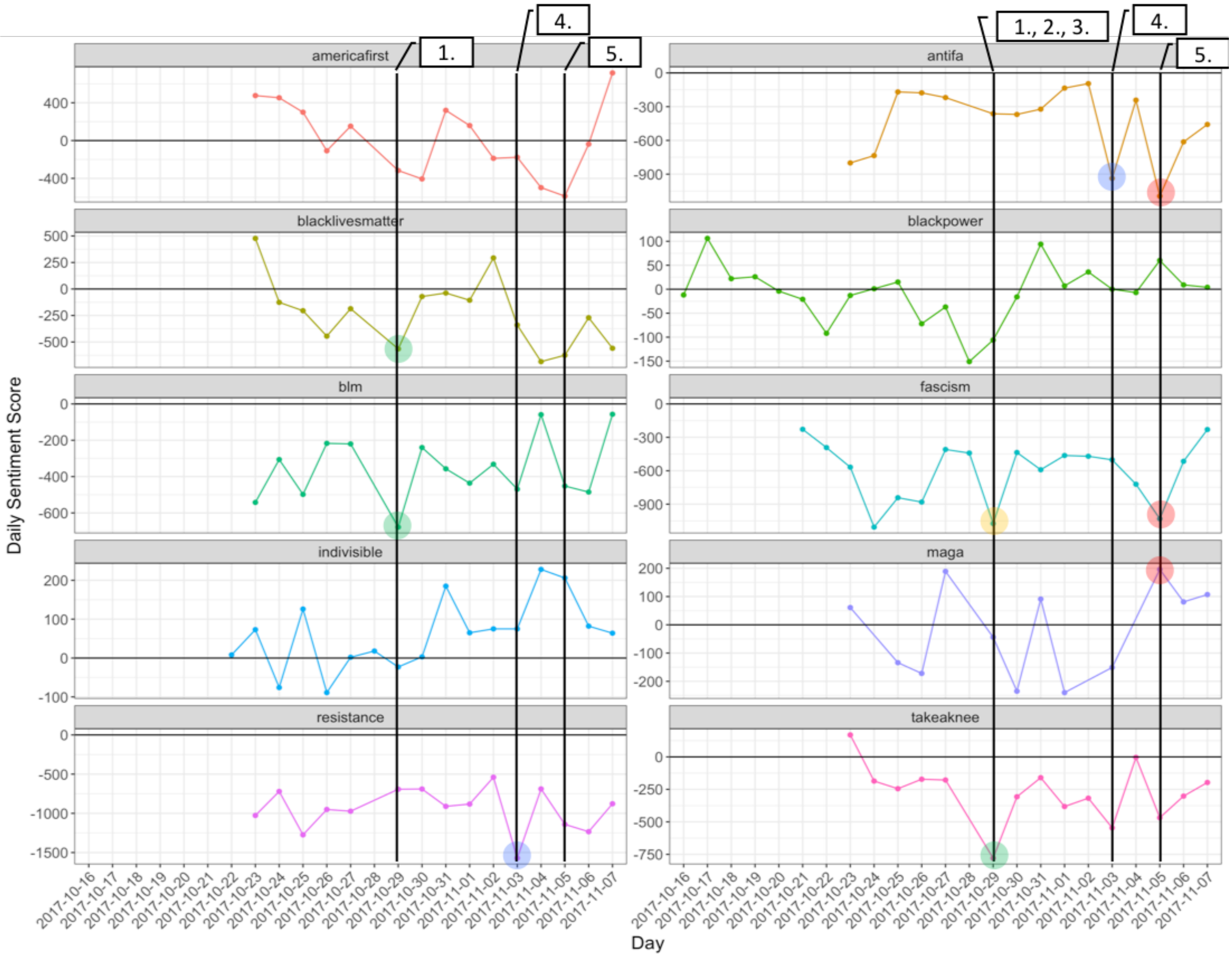
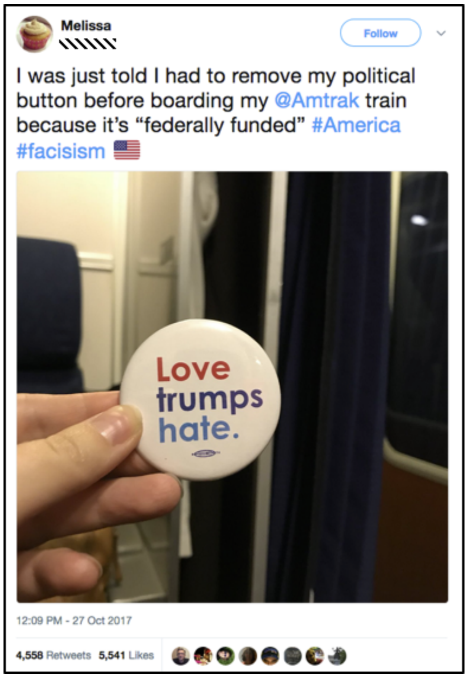


Figure 22. Protest Hashtag Time Series



Melissa

Follow

I was just told I had to remove my political button before boarding my @Amtrak train because it's "federally funded" #America #facisism 🇺🇸

12:09 PM - 27 Oct 2017

4,558 Retweets 5,541 Likes

Figure 23. Amtrak - Love Trumps Hate Political Button

4.3.3 Topic Analysis Sentiment Analysis

The LDA tunning algorithm was used again on the Protest dataset and resulted in the plot in Figure 24.

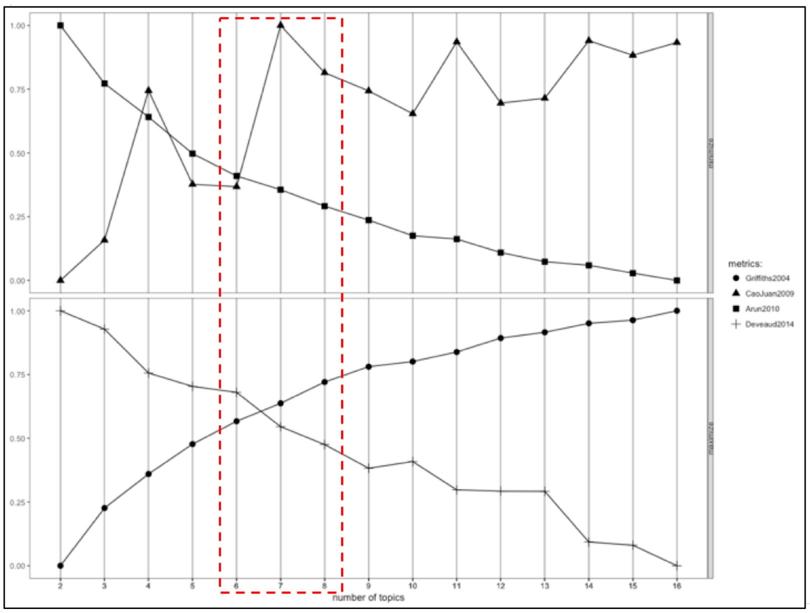


Figure 24. Protest LDA Tuning Plot

Interpreting the results from the LDA tunning graph in Figure 24 is challenging because of the of the observed divergence between the methods. The LDA algorithm was run with clusters = 6, 7, and 8 then manually compared to each other.

For the purpose of the Protest Bucket, clusters = 7 was chosen, as it appeared to have a balance between the number of topics and overlap between the topics, and can be seen in Table 16. Now that 7 topics were selected, the topics were renamed to assist the analyst in understanding what the topics include and can be seen in Table 16.

The Topics sentiment distributions can be found in the Figure 25. In the Violin Plot. From Figure 25 all topics were classified negative and can be found in Table 17.

From Table 17, all topics are classified as negative. The results of the time series analysis can be observed in Figure 26. For example:

1. 24OCT17: Lively Twitter discussion pertaining to free speech and chatter about the Antifa protests planned for 04NOV17. Which correspond to the *antifa, maga, politics*, and *protest* topics.

2. 29OCT17: Blacklivesmatter and Whitelivesmatter retweet in addition to a large discussion about illegals within the politics topics. Additionally a large number of tweets concerning ACLU support for kneeling NFL players and pro or against

Table 16. Protest Bucket Modeled with 7 Topics

Number	politics	resistance	maga	antifa	protest	blm	nfl
1	gop	resistance	maga	antifa	blm	blacklivesmatter	takeaknee
2	support	fascism	americafirst	fascism	trump	amp	indivisible
3	americafirst	blm	realdonaldtrump	trump	tcot	black	resist
4	vote	people	potus	retweet	2	blackpower	nfl
5	time	dont	trumptrain	realjameswoods	police	im	theresistance
6	amp	amp	america	http	takeaknee	american	trump
7	ht	youre	loudobbs	ad	protest	soros	stand
8	left	day	president	nov	racism	white	flag
9	americans	investigation	dnc	htt	tc	party	tco
10	tax	real	white	nytimes	join	stop	p2

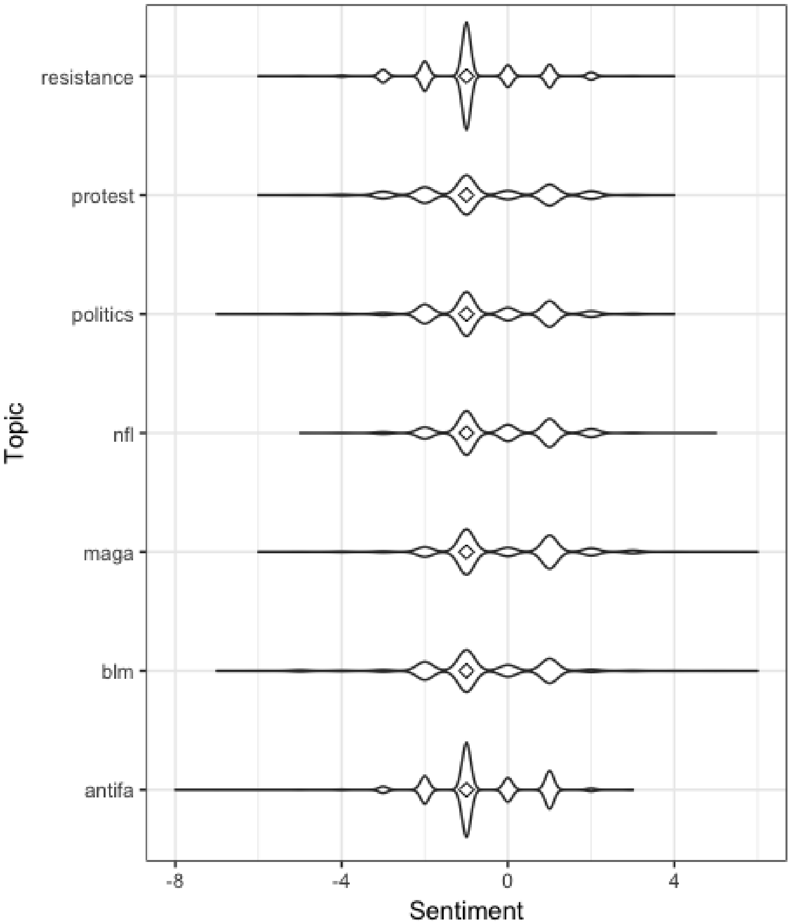


Figure 25. Protest Topic Violin Plot

Table 17. Protest Topic Classification

Negative	Neutral	Positive
resistance		
protest		
politics		
nfl		
maga		
blm		
antifa		

resistance type tweets. Furthermore, NFL Protests reached A boiling point, as players, owners cancel social concerns meeting[14]. Which all pertain to *politics,*

[14] http://deadline.com/2017/10/nfl-anthem-protests-reach-boiling-point-1202197034/

protest and *nfl* topics.

3. 03NOV2017: Antifa Civil War on November 4 Was Really Just a Few Protests Against Trump[15]. Which corresponds to the *antifa, politics*, and *resistance* topics.

4. 04NOV17: Significant portion of tweets “making-fun-of” Antifa and resistance fighters for a limited amount of protests on 04NOV17 which had been declared the start of the Antifa Civil war.

[15] `http://www.newsweek.com/antifa-civil-war-november-4-really-just-few-protests-against-trump-702150`

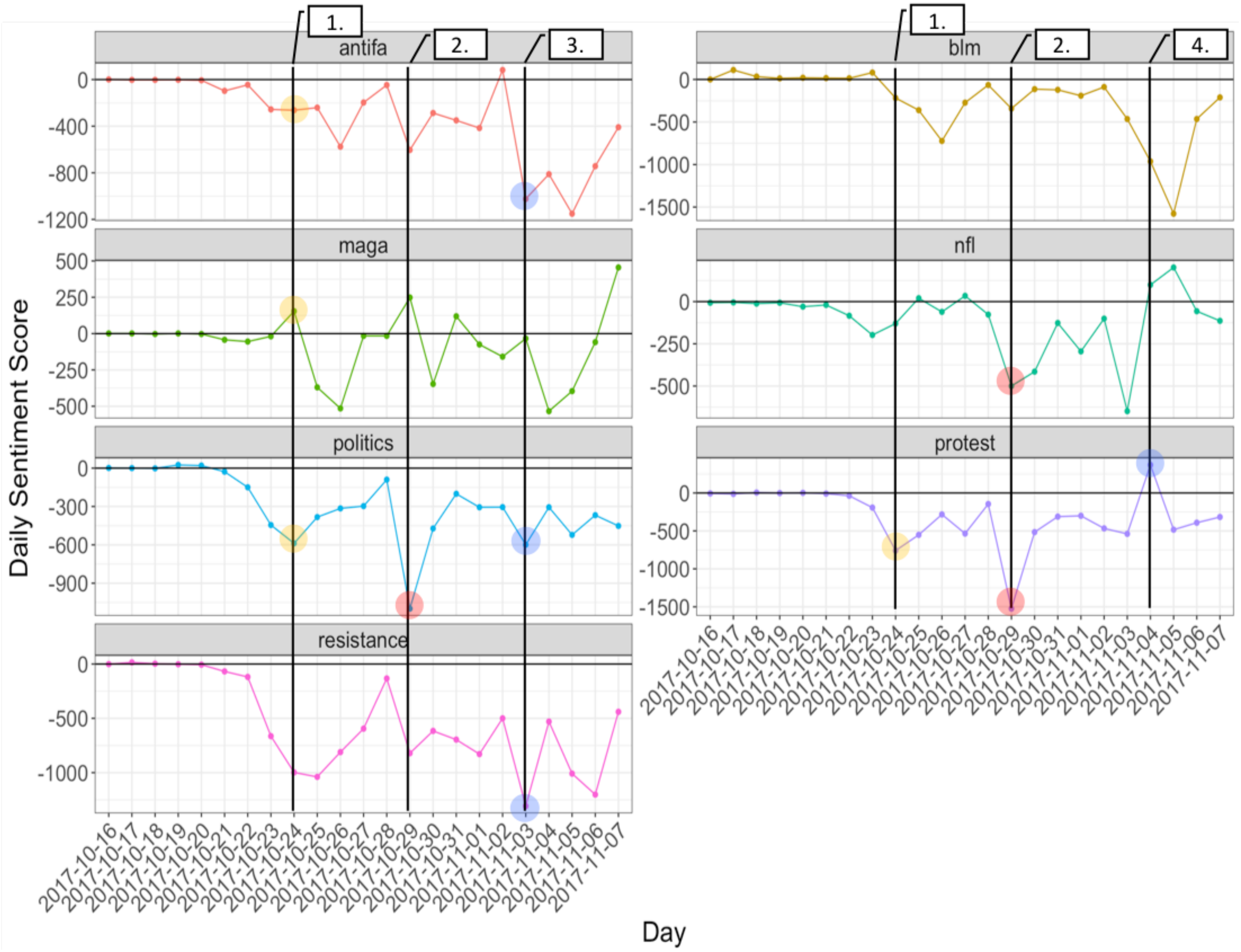


Figure 26. Protest Topic Time Series

4.4 Polling Comparison

The North Korea bucket was comprised of 11 different #hashtags and was also contained 7 different topics and the Protest bucket contained 10 different #hashtags and 7 topics. Generally, any lens the North bucket was looked at, showed the country in a negative light. With the primary exception being sanctions were being discussed. The Protest bucket was negative in every aspect.

In itself, tweets contain an immense amount of insight. Tweets are raw statements at best or near-gibberish at its worst. An average of ≈ 9 words were written together in an attempt to make a point or just let their voice be heard. The entire time their underlying sentiment and opinion could be gleaned.

The #hashtags and topics used in the analysis lead to an interesting array of hot-button issues throughout the world. Previously, the sentiment of Twitter data has been calculated and plotted through time to show the sentiment change of a #hashtag or topic. While the algorithm has been tested and shown to classify with accuracies between 72% - 82%, an additional validation method would be to compare polling results from an independent organization.

The Pew Research Center is a nonpartisan organization that aims to inform the public about issues, attitudes, and popular trends. The research center conducts nonpartisan public opinion polling, and other data-driven social science research [51]. Additionally, the Gallop is another well regarded nonpartisan organization that utilizes analytics to help leaders and organizations identify opportunities to create change. One of the ways in which Gallop does this is through polling individuals [52].

Gallup and Pew both conduct interviews with polling respondents to determine a response to a question, which takes time. For example, it took Gallup 52 days (16 June 2016 to 08 August 2016) for its interviewers to ask five questions to $\approx 26,000$ respondents [53]. However, in this analysis 395,063 tweets were collected over 16 days (23 October 2017 to 07 November 2017) and could produce sentiments within minutes of receiving the final Twitter data.

In a Pew Research Center article, respondents were asked about their views towards North Korea and then further asked about their opinion towards sanctions directed at the country. Respondents in the article reported they had a "Very unfavorable view of North Korea" 61% of the time. Additionally, 61% believed that sanctions were the preferable course of action instead of attempts to build closer ties with the country [54]. When the North Korea Bucket is looked at as a whole, it reveals that 60% of tweets show a negative sentiment. Furthermore, when tweets concerning #sanctions are looked at, those tweets show 64% of the tweets have a positive sentiment. Finally, when the topic *sanctions* is investigated, 57% of the tweets again have a positive sentiment. These results show that Twitter is generally in line with the results of the large polling organizations.

In a Gallup article, the percentage of U.S. adults who were fans of professional football fell from 67% in December 2012, to 57% in October 2017 [55]. The comparison to the Twitter data is much harder in this example because no #hashtags were specifically acquired for the National Football League (NFL). The tweets reveal that the *nfl* topic, has a 39% positive sentiment.

While the Twitter results are similar to the Pew and Gallup results, the twitter results cannot be confused or used with the same statistical confidence. The Twitter results in this report are not a proper random sample of all the tweets. However, the power and usefulness of the Twitter analysis is in how quick an analysis can be done and in remarkably how similar the results compare to very deliberate investigations completed by Pew and Gallup.

V. Conclusions and Future Research

5.1 Conclusion

Social media data, and in this case Twitter data is growing at an incredible rate. This rate is only expected to increase in the coming years. Getting a handle on the data now and determining ways to develop insights into that data is of critical importance. During the course of this paper, the sentiment of Twitter data was determined in one of two ways. Either by using specific #hashtags of interest or by using a Topic Modeling method to determine the underlying topics present within a dataset. The topic modeling approach could prove incredibly useful and powerful when dealing with large datasets not developed from #hashtags, when acquiring tweets during a specific time period or geographical location or if tweets are gathered in some other fashion.

5.1.1 North Korea

Based on the observed data and results found in Section 4.2.2 the sentiment of Twitter data is easily determined within the North Korea Bucket of data. Which leads to the conclusion that the sentiment associated with North Korea is negative. While this does not appear to be an incredible statement to any reader that keeps apprised of current events across the globe. It is incredibly useful when investigating new and unknown topics. Furthermore, the analysis of sentiment was critical in seeing how the sentiment changed through time which provided real-world insight into events happening worldwide. For example, the North Korea Nuclear Testing area tunnel collapse very clearly showed up in both the #hashtag and topic analysis. Interestingly in the North Korea bucket, most of the events that were discovered within the time series analysis were very easily confirmed with a quick Internet news search. This is most likely due to the lack of Twitter in North Korea and many of the events make it to mainstream news outlets. The methodology presented shows how that from start to finish an analyst can explore an unknown dataset, gain an in-depth understanding of the data quickly and easily and then

5.1.2 Protests

The analysis of the Protest bucket holds immense possibilities. Which leads to the conclusion that the sentiment associated with Protest is negative. Where the North Korea dataset is much easier to understand because many of the events are very easily found through mainstream news outlets, the Protest data is much more convoluted with multiple subgroups. In Section 4.3.1.1 N-Grams were investigated. Originally it was believed that the cleaning algorithm was improperly cleaning the tweets because words like: *ctl*, and *fbr* appeared. Upon further investigation these were additional #hashtags (#ctl and #fbr, which are: *ConnectTheLeft* and *FollowBackResistance*) that subgroups utilized to somewhat discretely share articles, tweets, and concepts that were interesting to this group of people. During the time series analysis, many of the large changes in sentiment were combinations of much smaller events that when combined resulted in a pronounced change in sentiment. Furthermore, because many of the discussed topics were smaller, they did not make mainstream news sources and were primarily only discussed on Twitter. Utilizing sentiment additional exploration into some of the different Protest #hashtags or topics could lead to a better understanding of the beliefs, organization, content and upcoming actions associated with these groups.

5.1.3 Polling Comparison

Finally, when the sentiment data was compared to real-world polling events, similar results were observed between the two. Of note, the Twitter data used was only a small sample of English tweets. With a small sample it is challenging to make a strong statement against the polling activities done by Pew and Gallup, however, this analysis has shown interesting similarities between polling and Twitter sentiment analysis. The level of approval of interviewees about North Korea was 61% and the calculated value using Twitter sentiment was 60%. When asked about sanctions the polling response was 61% for additional sanctions and the Twitter data showed that 64% of the sentiment was positive for the #sacntions and 57% positive when looking at the *sanctions* topic.

While not as impressive as the North Korea results. The Protest data did reveal that the **nfl** topic only had 39% positive sentiment, as opposed to the polling values of 57% of people who would still consider themselves fans of the NFL.

The computation of Twitter sentiment opinion is not the same as directly asking interviewees a number of questions. It is still powerful in the fact that the Twitter sentiment was collected discretely and very quickly and is able to provide a quick insight for decision makers who may not have the luxury of a long and detailed analysis of peoples opinion.

5.2 Future Research

Future research within this Sentiment Analysis can be broken down into two general concepts: Sentiment Determination and Topic Analysis.

5.2.1 Sentiment Determination

Within the Sentiment Determination future research area, incorporating Emojis, accounting for sarcasm and developing Machine Learning techniques instead of relying on lexicon dictionaries are areas that were not investigated in this report and would be very important areas to improve sentiment classification.

During this analysis, when the data is transformed into a Tidy format, all emojis are removed because there is a challenge in converting the emoji Unicode into a format that R can interpret and manipulate. Within the North Korea dataset, $\approx 26,000$ emojis are present which represent $\approx 3.76\%$ of the total words within the dataset. However, the use of the emojis is a powerful tool used to typically emphasize a point. If emoji were accounted for in the analysis it would more than likely assist the sentiment classification in providing more accurate results and add additional magnitude to the $TweetSentimentScore$ that would further help differentiate the sentiment between tweets.

Additionally, sarcasm is a difficult aspect of speech to detect and account for. Currently, this analysis makes no effort to properly identify and account for sarcasm in this analysis.

Finally, this research was limited in its ability to classify sentiment because it relied on a lexicon dictionary to classify sentiment. The algorithm was tested and shown to classify with accuracies between 72% - 82%, however using a Machine Learning approach that was able to account for emojis and sarcasm would more than likely improve upon the methods utilized in this paper.

5.2.2 Topic Analysis

Topic analysis was only briefly investigated within this research. Further investigation into developing the proper number of topics to select and additional exploration into the LDA algorithm are needed.

In order for the LDA algorithm to function properly, a user must manually select how many topics the text is to be segmented into. In this analysis, an LDA tuning algorithm was used which compared the maximization and minimization of four different methods. Ideally, the number of clusters selected will be where the difference between the maximization and minimization has been reduced. In the case of both the North Korea and Protest datasets, a divergence between the methods was observed which brought into play the art of operations research and required an educated selection which may or may not have been the correct number of topics to choose. Therefore additional research into determining the optimal number of topics could be of great use for future research.

Finally, LDA is a complex method of topic analysis that has multiple areas within the algorithm to adjust and manipulate to fine tune a result. This research did not experiment with any of the additional tuning parameters, which could be used to improve the results of the topic analysis.

Appendix A. Analysis Functions R Code

The following code describes the functions created to conduct sentiment analysis. The following R packages were used during the analysis:

- *data.table* [56]
- *readxl* [57]
- *utils* [58]
- *readr* [59]
- *plyr* [60]
- *dplyr* [61]
- *tidytext* [62]
- *stringr* [63]
- *ggplot2* [64]
- *lubridate* [65]
- *topicmodels* [46]
- *ggraph* [66]
- *igraph* [67]
- *ldatuning* [41]
- *widyr* [68]

```
# Bing Lexicon ----------------------------------------------------------

Bing ← as.data.frame(get_sentiments("bing")) %>%
  plyr::rename(c("word" = "Token", "sentiment" = "Sentiment"))

# Tidy and Scores -------------------------------------------------------

# Function to Tidy Twitter Data and remove all emoticons and maintain actual tweet
TD.Tidy ← function(DataFrame) {
  reg_words ← "([^A-Za-z_\\d#@']|'(?![A-Za-z_\\d#@]))"

  TD_Tidy ← DataFrame %>%
    dplyr::mutate(cleantext = str_replace_all(text,
        "https://t.co/[A-Za-z\\d]+|http://[A-Za-z\\d]+|&amp;|&lt;|&gt;|RT|https", "")) %>%
    dplyr::mutate(cleantext = str_replace_all(cleantext, "#", "")) %>%
    dplyr::mutate(cleantext = str_replace_all(cleantext, "http", "")) %>%
    dplyr::mutate(cleantext = str_replace_all(cleantext, "RT", "")) %>%
                 # Remove retweet note
    dplyr::mutate(cleantext = str_replace_all(cleantext, "[:punct:]", "")) %>%
    dplyr::mutate(cleantext = str_replace_all(cleantext, "[^[:alnum:]///' ]", "")) %>%
                 # Remove Emojis
    tidytext::unnest_tokens(output = word,
                            input = cleantext,
                            token = "words",
                            drop = TRUE) %>%
    dplyr::filter(!word %in% stop_words$word) %>%
    plyr::rename(c("word" = "Token"))
  return(TD_Tidy)
}

# Function to Calculate Sentiment Scores that will acount for sentiment by hashtag or topic
# For HT_Topic select: "hashtag" or "topic"
TD.Scores ← function(DataFrameTidy, HT_Topic) {
  if(HT_Topic == "hashtag") {
    TD_Hashtag_Scores ← DataFrameTidy %>%
      dplyr::inner_join(Bing, by = "Token") %>%
      dplyr::mutate(method = "Bing") %>%
      dplyr::group_by(text, method, hashtag, created, key, Sentiment) %>%
      dplyr::count(method, hashtag, created, key, Sentiment) %>%
      tidyr::spread(Sentiment, n, fill = 0) %>%
      dplyr::mutate(TweetSentimentScore = positive - negative) %>%
      dplyr::mutate(TweetSentiment = ifelse(TweetSentimentScore == 0, "neutral",
                                     ifelse(TweetSentimentScore > 0, "positive", "negative"))) %>%
      dplyr::mutate(date = lubridate::as_date(created))
    return(TD_Hashtag_Scores)
  } else {
    TD_Topic_Scores ← DataFrameTidy %>%
      dplyr::inner_join(Bing, by = "Token") %>%
      dplyr::mutate(method = "Bing") %>%
      dplyr::group_by(text, method, Topic, created, key, Sentiment) %>%
      dplyr::count(method, Topic, created, key, Sentiment) %>%
      tidyr::spread(Sentiment, n, fill = 0) %>%
      dplyr::mutate(TweetSentimentScore = positive - negative) %>%
      dplyr::mutate(TweetSentiment = ifelse(TweetSentimentScore == 0, "neutral",
                                     ifelse(TweetSentimentScore > 0, "positive", "negative"))) %>%
      dplyr::mutate(date = lubridate::as_date(created))
    return(TD_Topic_Scores)
  }
}

# Function to merge terms within a dataframe
Merge.Terms ← function(DataFrame, term, term.replacement){
  for(i in 1: length(DataFrame$text)){
    DataFrame[i, "text"] ← DataFrame[i, "text"] %>%
      gsub(pattern=as.character(term),
           replacement=as.character(term.replacement),
           ignore.case = T)
  }
  DataFrame ← DataFrame
}

# LDA ---------------------------------------------------------------------

# LDA number of optimal clusters for a given dataset
# num_cores = 2L for dual core
Number.Topics ← function(DataFrame,
                          num_cores,
                          min_clusters = 2,
                          max_clusters = 12,
                            skip = 2,
                          set_seed = 1234) {
  lda_prep ← DataFrame %>%
    dplyr::mutate(text = iconv(DataFrame$text, ”latin1”, ”ASCII”, sub=””)) %>%
    dplyr::mutate(text = stringr::str_replace_all(text, ”#”, ””)) %>%
              # Remove hashtag
    dplyr::mutate(text = stringr::str_replace_all(text, ”[:punct:]”, ””)) %>%
              # Remove punctuation
    dplyr::mutate(text = stringr::str_replace_all(text, ”RT”, ””)) %>%
              # Remove retweet note
    dplyr::mutate(text = stringr::str_replace_all(text, ”&amp”, ””)) %>%
              # Remove Accelerated Mobile Pages (AMP) note
    dplyr::mutate(text = stringr::str_replace_all(text,
        ”https://t.co/[A–Za–z\\d]+|http://[A–Za–z\\d]+|&amp;|&lt;|&gt;|RT|https”, ””)) %>%
            # Remove links
    dplyr::group_by(key) %>%
    tidytext::unnest_tokens(word, text) %>%
    dplyr::anti_join(stop_words) %>%
    dplyr::count(key, word, sort = TRUE) %>%
    tidytext::cast_dtm(key, word, n) # create DTM

  # Compute Values
  values ← ldatuning::FindTopicsNumber(lda_prep,
                         topics = seq(from = min_clusters,
                         to = max_clusters,
                         by = skip),
                         metrics = c(”Griffiths2004”, ”CaoJuan2009”, ”Arun2010”, ”Deveaud2014”),
                         method = ”Gibbs”,
                         control = list(seed = set_seed),
                         mc.cores = num_cores,
                         verbose = TRUE)

  # Plot
  columns ← base::subset(values, select = 2:ncol(values))
  values ← base::data.frame(values[”topics”], base::apply(columns, 2, function(column)
                                {scales::rescale(column, to = c(0, 1), from = range(column))}))
  values ← reshape2::melt(values, id.vars = ”topics”, na.rm = TRUE)
  values$group ← values$variable %in% c(”Griffiths2004”, ”Deveaud2014”)
  values$group ← base::factor(values$group,
                               levels = c(FALSE, TRUE),
                                 labels = c("minimize", "maximize"))
  p ← ggplot(values, aes_string(x = "topics", y = "value", group = "variable"))
  p ← p + geom_line()
  p ← p + geom_point(aes_string(shape = "variable"), size = 3)
  p ← p + guides(size = FALSE, shape = guide_legend(title = "metrics:"))
  p ← p + scale_x_continuous(breaks = values$topics)
  p ← p + labs(x = "number of topics", y = NULL)
  p ← p + facet_grid(group ~ .)
  p ← p + theme_bw() %+replace% theme(panel.grid.major.y = element_blank(),
                                      panel.grid.minor.y = element_blank(),
                                      panel.grid.major.x = element_line(colour = "grey70"),
                                      panel.grid.minor.x = element_blank(),
                                      legend.key = element_blank(),
                                      strip.text.y = element_text(angle = 90))
}


# Prepare tweet text, create DTM, conduct LDA, display data terms associated with each topic,
# Assign topic to tweet
Tweet.Topics ← function(DataFrame,
                        clusters,
                        method = "Gibbs",
                        seed = 1234,
                        num_terms = 10) {
  lda_prep ← DataFrame %>%
    dplyr::mutate(text = iconv(DataFrame$text, "latin1", "ASCII", sub="")) %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "#", "")) %>%
              # Remove hashtag
    dplyr::mutate(text = stringr::str_replace_all(text, "[:punct:]", "")) %>%
              # Remove punctuation
    dplyr::mutate(text = stringr::str_replace_all(text, "RT", "")) %>%
              # Remove retweet note
    dplyr::mutate(text = stringr::str_replace_all(text, "&amp", "")) %>%
              # Remove Accelerated Mobile Pages (AMP) note
    dplyr::mutate(text = stringr::str_replace_all(text,
        "https://t.co/[A-Za-z\\d]+|http://[A-Za-z\\d]+|&amp;|&lt;|&gt;|RT|https", "")) %>%
          # Remove links
    dplyr::group_by(key) %>%
    tidytext::unnest_tokens(word, text) %>%
    dplyr::anti_join(stop_words) %>%
    dplyr::count(key, word, sort = TRUE) %>%
    tidytext::cast_dtm(key, word, n)

  # Run LDA using Gibbs sampling
  ldaout ← LDA(lda_prep, k = clusters, method = method, control = list(seed = seed))

  ldaout_topics ← as.matrix(topicmodels::topics(ldaout))

  ldaout_terms ← as.matrix(topicmodels::terms(ldaout, num_terms))

  # probabilities associated with each topic assignment
  topicProbabilities ← as.data.frame(ldaout@gamma)
  data.topics ← topics(ldaout, 1)
  data.terms ← as.data.frame(terms(ldaout, num_terms), stringsAsFactors = FALSE)
  print(data.terms)
  View(data.terms)

  # Creates a dataframe to store the Lesson Number and the most likely topic
  tweettopics.df ← as.data.frame(data.topics)
  tweettopics.df ← dplyr::transmute(tweettopics.df,
                                    LessonId = rownames(tweettopics.df),
                                    Topic = data.topics)
  tweettopics.df$ArticleNo ← as.character(tweettopics.df$LessonId)

  # Clean up and rename coluns to match previous dataframes
  tweettopics ← tweettopics.df %>%
    dplyr::select(c(”ArticleNo”, ”Topic”)) %>%
    plyr::rename(c(”ArticleNo” = ”key”))

  # Join original twitter data frame with tweet topics
  Tweet.Topics ← dplyr::inner_join(DataFrame, tweettopics, by = ”key”)

  return(Tweet.Topics)
}

# Min / Max Scores ------------------------------------------------------------

# Output Minimum scores. NULL will find min across entire dataframe.
# Or select by hashtag or topic.
# For HT_Topic select: ”hashtag” or ”topic”
```

```
TD.Min.Scores ← function(DataFrameTidyScores, HT_Topic, HT_Topic_Seletion = NULL) {
  if(HT_Topic == ”hashtag” & is.null(HT_Topic_Seletion)) {
    TD_HT_noSel_Min_Scores ← DataFrameTidyScores %>%
      dplyr::arrange(TweetSentimentScore) %>%
      head()
    return(TD_HT_noSel_Min_Scores)
  } else if(HT_Topic == ”hashtag” & !is.null(HT_Topic_Seletion)) {
    TD_HT_Sel_Min_Scores ← DataFrameTidyScores %>%
      dplyr::filter(hashtag == HT_Topic_Seletion) %>%
      dplyr::arrange(TweetSentimentScore) %>%
      head()
    return(TD_HT_Sel_Min_Scores)
  } else if(HT_Topic == ”topic” & is.null(HT_Topic_Seletion)) {
    TD_Topic_noSel_Min_Scores ← DataFrameTidyScores %>%
      dplyr::arrange(TweetSentimentScore) %>%
      head()
    return(TD_Topic_noSel_Min_Scores)
  } else {
    TD_Topic_Sel_Min_Scores ← DataFrameTidyScores %>%
      dplyr::filter(Topic == HT_Topic_Seletion) %>%
      dplyr::arrange(TweetSentimentScore) %>%
      head()
    return(TD_Topic_Sel_Min_Scores)
  }
}


# Output Maximum scores.
# NULL will find min across entire dataframe.
# Or select by hashtag or topic.
# For HT_Topic select: ”hashtag” or ”topic”
TD.Max.Scores ← function(DataFrameTidyScores, HT_Topic, HT_Topic_Seletion = NULL) {
  if(HT_Topic == ”hashtag” & is.null(HT_Topic_Seletion)) {
    TD_HT_noSel_Max_Scores ← DataFrameTidyScores %>%
      dplyr::arrange(−TweetSentimentScore) %>%
      head()
    return(TD_HT_noSel_Max_Scores)
  } else if(HT_Topic == ”hashtag” & !is.null(HT_Topic_Seletion)) {
    TD_HT_Sel_Max_Scores ← DataFrameTidyScores %>%
      dplyr::filter(hashtag == HT_Topic_Seletion) %>%
      dplyr::arrange(−TweetSentimentScore) %>%
      head()
    return(TD_HT_Sel_Max_Scores)
  } else if(HT_Topic == ”topic” & is.null(HT_Topic_Seletion)) {
    TD_Topic_noSel_Max_Scores ← DataFrameTidyScores %>%
      dplyr::arrange(−TweetSentimentScore) %>%
      head()
    return(TD_Topic_noSel_Max_Scores)
  } else {
    TD_Topic_Sel_Max_Scores ← DataFrameTidyScores %>%
      dplyr::filter(Topic == HT_Topic_Seletion) %>%
      dplyr::arrange(−TweetSentimentScore) %>%
      head()
    return(TD_Topic_Sel_Max_Scores)
  }
}

# Word Grams −−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−−

# Most common positive and negative words
TD.PosNeg.Words ← function(DataFrameTidy, Lexicon = ”Bing”, filterword = NULL) {
  TD_PosNeg_Words ← DataFrameTidy %>%
    dplyr::inner_join(eval(as.name(Lexicon)), by = ”Token”) %>%
    dplyr::filter(!(Token %in% filterword)) %>%
    dplyr::count(Token, Sentiment) %>%
    dplyr::ungroup() %>%
    dplyr::group_by(Sentiment) %>%
    dplyr::top_n(10, n) %>%
    dplyr::ungroup() %>%
    dplyr::mutate(Token = reorder(Token, n)) %>%
    ggplot2::ggplot(aes(Token, n, fill = Sentiment)) +
    geom_col(show.legend = FALSE) +
    facet_wrap(~Sentiment, scales = ”free_y”) +
    labs(y = ”Count”,
          x = NULL) +
    ggtitle('Most common positive and negative words utilizing the Bing Lexicon') +
    coord_flip()
  return(TD_PosNeg_Words)
}

# Uni−Gram
```

```
TD.Unigram ← function(DataFrame){
  TD_Unigram ← DataFrame %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "RT", "")) %>%
              # Remove retweet note
    dplyr::mutate(text = stringr::str_replace_all(text, "&amp", "")) %>%
              # Remove Accelerated Mobile Pages (AMP) note
    dplyr::mutate(text = stringr::str_replace_all(text,
       "https://t.co/[A-Za-z\\d]+|http://[A-Za-z\\d]+|&amp;|&lt;|&gt;|RT|https", "")) %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "#", "")) %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "[:punct:]", "")) %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "[^[:alnum:]///' ]", "")) %>%
              # Remove Emojis
    tidytext::unnest_tokens(word, text) %>%
    dplyr::filter(!word %in% c(stop_words$word, '[0-9]+')) %>%
    dplyr::count(word, sort = TRUE)
}

# Bi-Gram
TD.Bigram ← function(DataFrame){
  TD_Bigram ← DataFrame %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "RT", "")) %>%
              # Remove retweet note
    dplyr::mutate(text = stringr::str_replace_all(text, "&amp", "")) %>%
              # Remove Accelerated Mobile Pages (AMP) note
    dplyr::mutate(text = stringr::str_replace_all(text,
       "https://t.co/[A-Za-z\\d]+|http://[A-Za-z\\d]+|&amp;|&lt;|&gt;|RT|https", "")) %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "#", "")) %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "[:punct:]", "")) %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "[^[:alnum:]///' ]", "")) %>%
              # Remove Emojis
    tidytext::unnest_tokens(bigram, text, token = "ngrams", n=2) %>%
    tidyr::separate(bigram, c("word1", "word2"), sep = " ") %>%
    dplyr::filter(!word1 %in% c(stop_words$word, '[0-9]+')) %>%
    dplyr::filter(!word2 %in% c(stop_words$word, '[0-9]+')) %>%
    dplyr::count(word1, word2, sort = TRUE)
}

# Tri-Gram
TD.Trigram ← function(DataFrame) {
  TD_Trigram ← DataFrame %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "RT", "")) %>%
              # Remove retweet note
    dplyr::mutate(text = stringr::str_replace_all(text, "&amp", "")) %>%
              # Remove Accelerated Mobile Pages (AMP) note
    dplyr::mutate(text = stringr::str_replace_all(text,
       "https://t.co/[A-Za-z\\d]+|http://[A-Za-z\\d]+|&amp;|&lt;|&gt;|RT|https", "")) %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "#", "")) %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "[:punct:]", "")) %>%
    dplyr::mutate(text = stringr::str_replace_all(text, "[^[:alnum:]///' ]", "")) %>%
              # Remove Emojis
    tidytext::unnest_tokens(trigram, text, token = "ngrams", n=3) %>%
    tidyr::separate(trigram, c("word1", "word2", "word3"), sep = " ") %>%
    dplyr::filter(!word1 %in% c(stop_words$word, '[0-9]+')) %>%
    dplyr::filter(!word2 %in% c(stop_words$word, '[0-9]+')) %>%
    dplyr::filter(!word3 %in% c(stop_words$word, '[0-9]+')) %>%
    dplyr::count(word1, word2, word3, sort = TRUE)
}

# Bi-Gram Network
# acceptable Layouts:  'star', 'circle', 'gem', 'dh', 'graphopt', 'grid', 'mds',
# 'randomly', 'fr', 'kk', 'drl', 'lgl'
TD.Bigram.Network ← function(BiGramDataFrame,
                             number = 300,
                             layout = "fr",
                             edge_color = "royalblue",
                             node_color = "black", n
                             ode_size = 3,
                             seed = 1234) {
  TD_Bigram_Network ← BiGramDataFrame %>%
    dplyr::filter(n > number) %>%
    igraph::graph_from_data_frame()

  set.seed(seed)

  ggraph::ggraph(TD_Bigram_Network, layout = layout) +
    geom_edge_link(aes(edge_alpha = n,
                       edge_width = n),
                       edge_colour = edge_color,
                       show.legend = TRUE,
                       end_cap = circle(.07, 'inches')) +
    geom_node_point(colour = node_color, size = node_size) +
    geom_node_text(aes(label = name), vjust = 1, hjust = 1, repel = TRUE) +
    ggtitle("Bi-Gram Network") +
    theme_void()
}

# Word Correlations -----------------------------------------------------

# Word Correlations
TD.Word.Corr ← function(DataFrameTidy, n, sort = TRUE) {
  TD_Word_Correlation ← DataFrameTidy %>%
    dplyr::group_by(Token) %>%
    dplyr::filter(n() >= n) %>%
    widyr::pairwise_cor(Token, key, sort = sort)
}

# Word Correlations Plot
# acceptable Layouts:  'star', 'circle', 'gem', 'dh', 'graphopt', 'grid', 'mds',
# 'randomly', 'fr', 'kk', 'drl', 'lgl'
TD.Word.Corr.Plot ← function(WordCorr,
                              Correlation = 0.15,
                              layout = "fr",
                              edge_color = "royalblue",
                              node_color = "black",
                              node_size = 2,
                              seed = 1234) {
  set.seed(seed)

  WordCorr %>%
    filter(correlation > Correlation) %>%
    graph_from_data_frame() %>%
    ggraph::ggraph(layout = layout) +
    geom_edge_link(aes(edge_alpha = correlation, edge_width = correlation),
                       edge_colour = edge_color,
                       show.legend = TRUE) +
    geom_node_point(colour = node_color, size = node_size) +
    geom_node_text(aes(label = name), repel = TRUE) +
    ggtitle("Word Correlation Network") +
    theme_void()
}
```

```
# Sentiment Distributions -----------------------------------------------

# TweetSentiment Corpus Distribution
TD.Corups.Distribution ← function(DataFrameTidyScores,
                                  binwidth = 1,
                                  colour = ”black”,
                                  fill = ”white”) {
  TD_Corups_Distribution ← DataFrameTidyScores %>%
    ggplot2::ggplot(aes(TweetSentimentScore)) +
    geom_histogram(binwidth = binwidth, colour = colour, fill = fill) +
    theme(legend.position = ”none”) +
    ggtitle(”Sentiment Score Distribution”) +
    xlab(’Sentiment’) +
    ylab(’Count’)
  return(TD_Corups_Distribution)
}

# TweetSentiScore Distribution by each Hashtag or Topic
# For HT_Topic select: ”hashtag” or ”topic”
TD.Distribution ← function(DataFrameTidyScores,
                           HT_Topic,
                           binwidth = 1,
                           color = ”black”,
                           fill = ”white”) {
  if(HT_Topic == ”hashtag”) {
    TD_HT_Distribution ← DataFrameTidyScores %>%
      ggplot2::ggplot(aes(TweetSentimentScore)) +
      geom_histogram(binwidth = binwidth, colour = color, fill = fill) +
      facet_wrap(~hashtag, ncol = 2) +
      theme(legend.position = ”none”) +
      ggtitle(”Sentiment Score Distribution Across all #Hashtags”) +
      xlab(’Sentiment’) +
      ylab(’Count’)
    return(TD_HT_Distribution)
  } else {
    TD_Topic_Distribution ← DataFrameTidyScores %>%
      ggplot2::ggplot(aes(TweetSentimentScore)) +
      geom_histogram(binwidth = binwidth, colour = color, fill = fill) +
      facet_wrap(~Topic, ncol = 2) +
      theme(legend.position = "none") +
      ggtitle("Sentiment Score Distribution Across all Topics") +
      xlab('Sentiment') +
      ylab('Count')
    return(TD_Topic_Distribution)
  }
}

# Visualizations ---------------------------------------------------------

# Box Plot select between hashtag or topic
# For HT_Topic select: "hashtag" or "topic"
TD.BoxPlot ← function(DataFrameTidyScores, HT_Topic) {
  if(HT_Topic == "hashtag") {
    TD_HT_BoxPlot ← DataFrameTidyScores %>%
      ggplot2::ggplot(aes(hashtag, TweetSentimentScore)) +
      ggplot2::geom_boxplot() +
      theme(legend.position = "none") +
      ggtitle("Sentiment Scores Across each #Hashtag") +
      xlab('#Hashtag') +
      ylab('Sentiment') +
      coord_flip()
    return(TD_HT_BoxPlot)
  } else {
    TD_Topic_BoxPlot ← DataFrameTidyScores %>%
      ggplot2::ggplot(aes(Topic, TweetSentimentScore)) +
      ggplot2::geom_boxplot() +
      theme(legend.position = "none") +
      ggtitle("Sentiment Scores Across each Topic") +
      xlab('Topic') +
      ylab('Sentiment') +
      coord_flip()
    return(TD_Topic_BoxPlot)
  }
}

# Violin Plot
# For HT_Topic select: "hashtag" or "topic"
TD.ViolinPlot ← function(DataFrameTidyScores, HT_Topic) {
  if(HT_Topic == "hashtag") {
    TD_HT_ViolinPlot ← DataFrameTidyScores %>%
      ggplot2:: ggplot(aes(hashtag, TweetSentimentScore)) +
      geom_violin(scale = "area") +
      stat_summary(fun.y = median, geom = "point", shape = 23, size = 2) +
      ggtitle("Sentiment Scores Across each #Hashtag") +
      xlab('#Hashtag') +
      ylab('Sentiment') +
      coord_flip()
    return(TD_HT_ViolinPlot)
  } else{
    TD_Topic_ViolinPlot ← DataFrameTidyScores %>%
      ggplot2:: ggplot(aes(Topic, TweetSentimentScore)) +
      geom_violin(scale = "area") +
      stat_summary(fun.y = median, geom = "point", shape = 23, size = 2) +
      ggtitle("Sentiment Scores Across each Topic") +
      xlab('Topic') +
      ylab('Sentiment') +
      coord_flip()
    return(TD_Topic_ViolinPlot)
  }
}

# Sentiment Timescale facet wrap
# For HT_Topic select: "hashtag" or "topic"
TD.TimeScale ← function(DataFrameTidyScores, HT_Topic) {
  if(HT_Topic == "hashtag") {
    TD_HT_TimeScale ← DataFrameTidyScores %>%
      dplyr::group_by(hashtag, date) %>%
      dplyr::summarise(DayScore = sum(TweetSentimentScore)) %>%
      ggplot2::ggplot(aes(x = factor(date), y = DayScore, colour = hashtag)) +
      geom_point() +
      geom_path(aes(group=1)) +
      geom_hline(yintercept = 0, color = "black") +
      facet_wrap(~hashtag, ncol = 2, scales = "free_y") +
      theme(legend.position = "none") +
      ggtitle("Sentiment Scores Across all #Hashtags") +
      xlab('Day') +
      ylab('Daily Sentiment Score') +
      theme(axis.text.x = element_text(angle = 45, hjust = 1))
    return(TD_HT_TimeScale)
  } else {
    TD_Topic_TimeScale ← DataFrameTidyScores %>%
      dplyr::group_by(Topic, date) %>%
      dplyr::summarise(DayScore = sum(TweetSentimentScore)) %>%
      ggplot2::ggplot(aes(x = factor(date), y = DayScore, colour = Topic)) +
      geom_point() +
      geom_path(aes(group=1)) +
      geom_hline(yintercept = 0, color = ”black”) +
      facet_wrap(∼Topic, ncol = 2, scales = ”free_y”) +
      theme(legend.position = ”none”) +
      ggtitle(”Sentiment Scores Across all Topics”) +
      xlab(’Day’) +
      ylab(’Daily Sentiment Score’) +
      theme(axis.text.x = element_text(angle = 45, hjust = 1))
    return(TD_Topic_TimeScale)
  }
}

# World Map of Tweets by hashtag
# For HT_Topic select: ”hashtag” or ”topic”
TD.WorldMap ← function(DataFrame, HT_Topic) {
  if(HT_Topic == ”hashtag”) {
    TD_HT_WorldMap ← ggplot2::map_data(”world”) %>%
      ggplot2::ggplot() +
      geom_polygon(aes(x = long, y = lat, group = group), colour = ”black”, fill = ”white”) +
      geom_jitter(data = DataFrame,
                  aes(x = as.numeric(longitude),
                      y = as.numeric(latitude),
                      colour = hashtag)) +
      ggtitle(”World Map of Tweets”) +
      theme(legend.position = ”bottom”) +
      scale_fill_continuous(guide = guide_legend(title = NULL)) +
      coord_quickmap()
    return(TD_HT_WorldMap)
  } else {
    TD_Topic_WorldMap ← ggplot2::map_data(”world”) %>%
      ggplot2::ggplot() +
      geom_polygon(aes(x = long, y = lat, group = group), colour = ”black”, fill = ”white”) +
      geom_jitter(data = DataFrame,
                  aes(x = as.numeric(longitude),
                      y = as.numeric(latitude),
                      colour = Topic)) +
      ggtitle(”World Map of Tweets”) +
      theme(legend.position = ”bottom”) +
      scale_fill_continuous(guide = guide_legend(title = NULL)) +
      coord_quickmap()
    return(TD_Topic_WorldMap)
  }
}
```

Appendix B. North Korea R Code

The Following code utilizes the function described in Appendix A and uses them to determine the sentiment in the North Korea Dataset

```
set.seed(1234)

# Access Twitter API ------------------------------------------------------------

consumer_key ← "xxxxxxxxxxxxxxxxxx"
consumer_secret ← "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
access_token ← "xxxxxxxxxxxxxxxxxx−xxxxxxxxxxxxxxxxxx"
access_secret ← "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

ht ← c("#northkorea",
        "#nuke",
        "#dprk",
        "#rocketman",
        "#missile",
        "#sanctions",
        "#test",
        "#KimJongUn",
        "#southkorea",
        "#WWIII",
        "#ww3")

no.of.tweets ← 1000

# Scrape Data from Twitter API for each #hashtag --------------------------

twitter_data ← list()
for (i in ht) {
  twitter_data[[i]] ← twListToDF(searchTwitter(i, n = no.of.tweets, lang = "en"))%>%
    mutate(hashtag = substr(i, 2, nchar(i)))
}

twitter_data_171107_NorthKorea ← map_df(twitter_data,rbind)
save(twitter_data_171107_NorthKorea, file = "twitter_data_171107_NorthKorea.RData")
```

```
# Combine North Korea Data Files ----------------------------------------

# Load in previously saved North Korea .RData files
load(”twitter_data_171023_NorthKorea.RData”)
load(”twitter_data_171024_NorthKorea.RData”)
load(”twitter_data_171025_NorthKorea.RData”)
load(”twitter_data_171026_NorthKorea.RData”)
load(”twitter_data_171027_NorthKorea.RData”)
load(”twitter_data_171029_NorthKorea.RData”)
load(”twitter_data_171030_NorthKorea.RData”)
load(”twitter_data_171031_NorthKorea.RData”)
load(”twitter_data_171101_NorthKorea.RData”)
load(”twitter_data_171102_NorthKorea.RData”)
load(”twitter_data_171103_NorthKorea.RData”)
load(”twitter_data_171104_NorthKorea.RData”)
load(”twitter_data_171105_NorthKorea.RData”)
load(”twitter_data_171106_NorthKorea.RData”)
load(”twitter_data_171107_NorthKorea.RData”)

# Bind North Korea files into one Data Frame
TD_NK ← rbind(twitter_data_171023_NorthKorea,
              twitter_data_171024_NorthKorea,
              twitter_data_171025_NorthKorea,
              twitter_data_171026_NorthKorea,
              twitter_data_171027_NorthKorea,
              twitter_data_171029_NorthKorea,
              twitter_data_171030_NorthKorea,
              twitter_data_171031_NorthKorea,
              twitter_data_171101_NorthKorea,
              twitter_data_171102_NorthKorea,
              twitter_data_171103_NorthKorea,
              twitter_data_171104_NorthKorea,
              twitter_data_171105_NorthKorea,
              twitter_data_171106_NorthKorea,
              twitter_data_171107_NorthKorea) %>%
  dplyr::mutate(key = paste(screenName, created)) %>%
  distinct(key, .keep_all = TRUE) # filter out all duplicate tweets.
```

```
# Hashtag Sentiment Analysis ----------------------------------------------

#### Data Exploration ####

# Unigram
NK_HT_Unigram ← TD.Unigram(DataFrame = TD_NK)

# Bigram
NK_HT_Bigram ← TD.Bigram(DataFrame = TD_NK)

# Trigram
NK_HT_Trigram ← TD.Trigram(DataFrame = TD_NK)

#### Merge Terms ####

# Merge terms
TD_NK_Merge ← Merge.Terms(DataFrame = TD_NK, ”north korea”, ”north_korea”)
TD_NK_Merge ← Merge.Terms(DataFrame = TD_NK_Merge, ”south korea”, ”south_korea”)
TD_NK_Merge ← Merge.Terms(DataFrame = TD_NK_Merge, ”southkorea”, ”south_korea”)
TD_NK_Merge ← Merge.Terms(DataFrame = TD_NK_Merge, ”northkorea”, ”south_korea”)
TD_NK_Merge ← Merge.Terms(DataFrame = TD_NK_Merge, ”ballistic missile”, ”ballistic_missile”)
TD_NK_Merge ← Merge.Terms(DataFrame = TD_NK_Merge, ”nuclear test”, ”nuclear_test”)
TD_NK_Merge ← Merge.Terms(DataFrame = TD_NK_Merge, ”president trump”, ”pdjt”)

# Unigram
NK_HT_Unigram ← TD.Unigram(DataFrame = TD_NK_Merge)

# Bigram
NK_HT_Bigram ← TD.Bigram(DataFrame = TD_NK_Merge)

# Trigram
NK_HT_Trigram ← TD.Trigram(DataFrame = TD_NK_Merge)

# Bigram Network
NK_HT_Bigram_Network ← TD.Bigram.Network(BiGramDataFrame = NK_HT_Bigram, number = 400)

#### Tidy and Socres ####
TD_NK_HT_Tidy ← TD.Tidy(DataFrame = TD_NK_Merge)

NK_HT_Scores ← TD.Scores(DataFrameTidy = TD_NK_HT_Tidy, HT_Topic = ”hashtag”)
```

```
#### Word Grams ####
NK_HT_PosNeg ← TD.PosNeg.Words(DataFrameTidy = TD_NK_HT_Tidy)


# filter out "trump"
NK_HT_PosNeg_trump ← TD.PosNeg.Words(DataFrameTidy = TD_NK_HT_Tidy, filterword = "trump")


# Bigram Network
stop  ← TD.Bigram.Network(BiGramDataFrame = NK_HT_Bigram, number = 400)


#### Min / Max Scores ####
NK_HT_Min ← TD.Min.Scores(DataFrameTidyScores = NK_HT_Scores, HT_Topic = "hashtag")
NK_HT_Min_dprk ← TD.Min.Scores(DataFrameTidyScores = NK_HT_Scores,
                               HT_Topic = "hashtag",
                               HT_Topic_Seletion = "dprk")


NK_HT_Max ← TD.Max.Scores(DataFrameTidyScores = NK_HT_Scores, HT_Topic = "hashtag")
NK_HT_Max_dprk ← TD.Max.Scores(DataFrameTidyScores = NK_HT_Scores,
                               HT_Topic = "hashtag",
                               HT_Topic_Seletion = "dprk")


#### Word Correlations ####
NK_HT_Word_Corr ← TD.Word.Corr(DataFrameTidy = TD_NK_HT_Tidy, n = 1000)


NK_HT_Word_Corr_Plot ← TD.Word.Corr.Plot(WordCorr = NK_HT_Word_Corr,
                                         layout = "fr",
                                         Correlation = 0.1)


#### Sentiment Distributions ####
NK_HT_Corp_Dist ← TD.Corups.Distribution(DataFrameTidyScores = NK_HT_Scores)


NK_HT_Dist ← TD.Distribution(DataFrameTidyScores = NK_HT_Scores, HT_Topic = "hashtag")


#### Visualizations ####
NK_HT_Box ← TD.BoxPlot(DataFrameTidyScores = NK_HT_Scores, HT_Topic = "hashtag")


NK_HT_Violin ← TD.ViolinPlot(DataFrameTidyScores = NK_HT_Scores, HT_Topic = "hashtag")


NK_HT_Time ← TD.TimeScale(DataFrameTidyScores = NK_HT_Scores, HT_Topic = "hashtag")
```

```
NK_HT_Map ← TD.WorldMap(DataFrame = TD_NK_Merge, HT_Topic = ”hashtag”)

# LDA Sentiment Analysis ---------------------------------------------------

#### LDA ####

# Produce graph of optimal number of topics
NK_LDA_Topics_Plot ← Number.Topics(DataFrame = TD_NK_Merge,
                                   num_cores = 2L,
                                   min_clusters = 2,
                                   max_clusters = 16,
                                   skip = 1)

# Tweet Topics
TD_NK_LDA_6 ← Tweet.Topics(DataFrame = TD_NK_Merge, clusters = 6, num_terms = 10)
TD_NK_LDA_8 ← Tweet.Topics(DataFrame = TD_NK_Merge, clusters = 8, num_terms = 10)
TD_NK_LDA_16 ← Tweet.Topics(DataFrame = TD_NK_Merge, clusters = 16, num_terms = 10)

# Choose 7
TD_NK_LDA ← Tweet.Topics(DataFrame = TD_NK_Merge, clusters = 7, num_terms = 10)

# Rename Topics
TD_NK_LDA ← TD_NK_LDA %>%
  dplyr::mutate(Topic = stringr::str_replace_all(Topic, ”^1$”, ”test”)) %>%
  dplyr::mutate(Topic = stringr::str_replace_all(Topic, ”^2$”, ”missile”)) %>%
  dplyr::mutate(Topic = stringr::str_replace_all(Topic, ”^3$”, ”politics”)) %>%
  dplyr::mutate(Topic = stringr::str_replace_all(Topic, ”^4$”, ”southkorea”)) %>%
  dplyr::mutate(Topic = stringr::str_replace_all(Topic, ”^5$”, ”dogmeat”)) %>%
  dplyr::mutate(Topic = stringr::str_replace_all(Topic, ”^6$”, ”sanctions”)) %>%
  dplyr::mutate(Topic = stringr::str_replace_all(Topic, ”^7$”, ”nuke”))

TD_NK_LDA_Tidy ← TD.Tidy(DataFrame = TD_NK_LDA)

NK_LDA_Scores ← TD.Scores(DataFrameTidy = TD_NK_LDA_Tidy, HT_Topic = ”topic”)

#### Min / Max Scores ####
NK_LDA_Min ← TD.Min.Scores(DataFrameTidyScores = NK_LDA_Scores, HT_Topic = ”topic”)
NK_LDA_Min_sanctions ← TD.Min.Scores(DataFrameTidyScores = NK_LDA_Scores,
                                     HT_Topic = ”topic”,
                                     T_Topic_Seletion = ”sanctions”)
```

```
NK_LDA_Max ← TD.Max.Scores(DataFrameTidyScores = NK_LDA_Scores, HT_Topic = "topic")
NK_LDA_Max_msanctions ← TD.Max.Scores(DataFrameTidyScores = NK_LDA_Scores,
                                        HT_Topic = "topic",
                                        HT_Topic_Seletion = "sanctions")


#### Sentiment Distributions ####
NK_LDA_Corp_Dist ← TD.Corups.Distribution(DataFrameTidyScores = NK_LDA_Scores)

NK_LDA_Dist ← TD.Distribution(DataFrameTidyScores = NK_LDA_Scores, HT_Topic = "topic")


#### Visualizations ####
NK_LDA_Box ← TD.BoxPlot(DataFrameTidyScores = NK_LDA_Scores, HT_Topic = "topic")

NK_LDA_Violin ← TD.ViolinPlot(DataFrameTidyScores = NK_LDA_Scores, HT_Topic = "topic")

NK_LDA_Time ← TD.TimeScale(DataFrameTidyScores = NK_LDA_Scores, HT_Topic = "topic")

NK_LDA_Map ← TD.WorldMap(DataFrame = TD_NK_LDA, HT_Topic = "topic")
```

Appendix C. Protest R Code

The Following code utilizes the function described in Appendix A and uses them to determine the sentiment in the Protest Dataset

```
set.seed(1234)

# Access Twitter API ------------------------------------------------------------

consumer_key ← "xxxxxxxxxxxxxxxxxx"
consumer_secret ← "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
access_token ← "xxxxxxxxxxxxxxxxxx-xxxxxxxxxxxxxxxxxx"
access_secret ← "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)

ht ← c("#antifa",
        "#resistance",
        "#fascism",
        "#blm",
        "#blacklivesmatter",
        "#blackpower",
        "#takeaknee",
        "#indivisible",
        "#americafirst",
        "#maga")

no.of.tweets ← 1000

# Scrape Data from Twitter API for each #hashtag --------------------------

twitter_data ← list()
for (i in ht) {
  twitter_data[[i]] ← twListToDF(searchTwitter(i, n = no.of.tweets, lang = "en"))%>%
    mutate(hashtag = substr(i, 2, nchar(i)))
}

twitter_data_171107_Protest ← map_df(twitter_data,rbind)
save(twitter_data_171107_Protest, file = "twitter_data_171107_Protest.RData")
```

```
# Combine Protest Data Files ---------------------------------------

# Load in previously saved Protest .RData files
load(”twitter_data_171023_Protest.RData”)
load(”twitter_data_171024_Protest.RData”)
load(”twitter_data_171025_Protest.RData”)
load(”twitter_data_171026_Protest.RData”)
load(”twitter_data_171027_Protest.RData”)
load(”twitter_data_171029_Protest.RData”)
load(”twitter_data_171030_Protest.RData”)
load(”twitter_data_171031_Protest.RData”)
load(”twitter_data_171101_Protest.RData”)
load(”twitter_data_171102_Protest.RData”)
load(”twitter_data_171103_Protest.RData”)
load(”twitter_data_171104_Protest.RData”)
load(”twitter_data_171105_Protest.RData”)
load(”twitter_data_171106_Protest.RData”)
load(”twitter_data_171107_Protest.RData”)

# Bind North Korea files into one Data Frame
TD_PRO ← rbind(twitter_data_171023_Protest,
               twitter_data_171024_Protest,
               twitter_data_171025_Protest,
               twitter_data_171026_Protest,
               twitter_data_171027_Protest,
               twitter_data_171029_Protest,
               twitter_data_171030_Protest,
               twitter_data_171031_Protest,
               twitter_data_171101_Protest,
               twitter_data_171102_Protest,
               twitter_data_171103_Protest,
               twitter_data_171104_Protest,
               twitter_data_171105_Protest,
               twitter_data_171106_Protest,
               twitter_data_171107_Protest) %>%
  dplyr::mutate(created = lubridate::as_datetime(created)) %>%
  dplyr::mutate(key = paste(screenName, created)) %>%
  distinct(key, .keep_all = TRUE) # filter out all duplicate tweets.
```

```
# Hashtag Sentiment Analysis --------------------------------------------

#### Data Exploration ####

# Unigram
PRO_HT_Unigram ← TD.Unigram(DataFrame = TD_PRO)

# Bigram
PRO_HT_Bigram ← TD.Bigram(DataFrame = TD_PRO)

# Trigram
PRO_HT_Trigram ← TD.Trigram(DataFrame = TD_PRO)

# Bigram Network
PRO_HT_Bigram_Network ← TD.Bigram.Network(BiGramDataFrame = PRO_HT_Bigram, number = 400)

#### Tidy and Socres ####
TD_PRO_HT_Tidy ← TD.Tidy(DataFrame = TD_PRO)

PRO_HT_Scores ← TD.Scores(DataFrameTidy = TD_PRO_HT_Tidy, HT_Topic = ”hashtag”)

#### Word Grams ####
PRO_HT_PosNeg ← TD.PosNeg.Words(DataFrameTidy = TD_PRO_HT_Tidy)

# filter out ”trump”
PRO_HT_PosNeg_trump ← TD.PosNeg.Words(DataFrameTidy = TD_PRO_HT_Tidy, filterword = ”trump”)

# Bigram Network
PRO_HT_Bigram_Network ← TD.Bigram.Network(BiGramDataFrame = PRO_HT_Bigram, number = 500)

#### Min / Max Scores ####
PRO_HT_Min ← TD.Min.Scores(DataFrameTidyScores = PRO_HT_Scores, HT_Topic = ”hashtag”)
# PRO_HT_Min_dprk ← TD.Min.Scores(DataFrameTidyScores = PRO_HT_Scores,
 HT_Topic = ”hashtag”, HT_Topic_Seletion = ”dprk”)

PRO_HT_Max ← TD.Max.Scores(DataFrameTidyScores = PRO_HT_Scores, HT_Topic = ”hashtag”)
# PRO_HT_Max_dprk ← TD.Max.Scores(DataFrameTidyScores = PRO_HT_Scores,
 HT_Topic = ”hashtag”,HT_Topic_Seletion = ”dprk”)
```

```
#### Word Correlations ####
PRO_HT_Word_Corr ← TD.Word.Corr(DataFrameTidy = TD_PRO_HT_Tidy, n = 1100)

PRO_HT_Word_Corr_Plot ← TD.Word.Corr.Plot(WordCorr = PRO_HT_Word_Corr,
 layout = "fr", Correlation = 0.1)

#### Sentiment Distributions ####
PRO_HT_Corp_Dist ← TD.Corups.Distribution(DataFrameTidyScores = PRO_HT_Scores)

PRO_HT_Dist ← TD.Distribution(DataFrameTidyScores = PRO_HT_Scores, HT_Topic = "hashtag")

#### Visualizations ####
PRO_HT_Box ← TD.BoxPlot(DataFrameTidyScores = PRO_HT_Scores, HT_Topic = "hashtag")

PRO_HT_Violin ← TD.ViolinPlot(DataFrameTidyScores = PRO_HT_Scores, HT_Topic = "hashtag")

PRO_HT_Time ← TD.TimeScale(DataFrameTidyScores = PRO_HT_Scores, HT_Topic = "hashtag")

PRO_HT_Map ← TD.WorldMap(DataFrame = TD_PRO, HT_Topic = "hashtag")

# LDA Sentiment Analysis --------------------------------------------------

#### LDA ####

# Produce graph of optimal number of topics
PRO_LDA_Topics_Plot ← Number.Topics(DataFrame = TD_PRO,
 num_cores = 2L, min_clusters = 2, max_clusters = 16,  skip = 1)

# Tweet Topics
TD_PRO_LDA_6 ← Tweet.Topics(DataFrame = TD_PRO, clusters = 6, num_terms = 10)
TD_PRO_LDA_8 ← Tweet.Topics(DataFrame = TD_PRO, clusters = 8, num_terms = 10)
TD_PRO_LDA_16 ← Tweet.Topics(DataFrame = TD_PRO, clusters = 16, num_terms = 10)

# Choose 7
TD_PRO_LDA ← Tweet.Topics(DataFrame = TD_PRO, clusters = 7, num_terms = 10)

# Rename Topics
TD_PRO_LDA ← TD_PRO_LDA %>%
   dplyr::mutate(Topic = stringr::str_replace_all(Topic, "^1$", "politics")) %>%
   dplyr::mutate(Topic = stringr::str_replace_all(Topic, "^2$", "resistance")) %>%
  dplyr::mutate(Topic = stringr::str_replace_all(Topic, ”ˆ3$”, ”maga”)) %>%
  dplyr::mutate(Topic = stringr::str_replace_all(Topic, ”ˆ4$”, ”antifa”)) %>%
  dplyr::mutate(Topic = stringr::str_replace_all(Topic, ”ˆ5$”, ”protest”)) %>%
  dplyr::mutate(Topic = stringr::str_replace_all(Topic, ”ˆ6$”, ”blm”)) %>%
  dplyr::mutate(Topic = stringr::str_replace_all(Topic, ”ˆ7$”, ”nfl”))

TD_PRO_LDA_Tidy ← TD.Tidy(DataFrame = TD_PRO_LDA)

PRO_LDA_Scores ← TD.Scores(DataFrameTidy = TD_PRO_LDA_Tidy, HT_Topic = ”topic”)

#### Min / Max Scores ####
PRO_LDA_Min ← TD.Min.Scores(DataFrameTidyScores = PRO_LDA_Scores, HT_Topic = ”topic”)
PRO_LDA_Min_sanctions ← TD.Min.Scores(DataFrameTidyScores = PRO_LDA_Scores,
 HT_Topic = ”topic”, HT_Topic_Seletion = ”sanctions”)

PRO_LDA_Max ← TD.Max.Scores(DataFrameTidyScores = PRO_LDA_Scores, HT_Topic = ”topic”)
PRO_LDA_Max_msanctions ← TD.Max.Scores(DataFrameTidyScores = PRO_LDA_Scores,
 HT_Topic = ”topic”, HT_Topic_Seletion = ”sanctions”)

#### Sentiment Distributions ####
PRO_LDA_Corp_Dist ← TD.Corups.Distribution(DataFrameTidyScores = PRO_LDA_Scores)

PRO_LDA_Dist ← TD.Distribution(DataFrameTidyScores = PRO_LDA_Scores, HT_Topic = ”topic”)

#### Visualizations ####
PRO_LDA_Box ← TD.BoxPlot(DataFrameTidyScores = PRO_LDA_Scores, HT_Topic = ”topic”)

PRO_LDA_Violin ← TD.ViolinPlot(DataFrameTidyScores = PRO_LDA_Scores, HT_Topic = ”topic”)

PRO_LDA_Time ← TD.TimeScale(DataFrameTidyScores = PRO_LDA_Scores, HT_Topic = ”topic”)

PRO_LDA_Map ← TD.WorldMap(DataFrame = TD_PRO_LDA, HT_Topic = ”topic”)


 j
```

Sentiment Analysis of Twitter Data

Problem Statement

Harvested Twitter data, analyzed for opinions and sentiment can provide powerful insight into a population. This insight can assist companies by letting them better understand their target population. Data was acquired through the Public Twitter Application Programming Interface (API). A methodology was developed that utilized a topic modeling and lexicographical approach to analyze the sentiment and opinions of text in English to determine a general sentiment.

Twitter Data Preparation

Cleaning:

- Remove "@" symbol
- Remove "#" symbol
- Remove "RT" symbol
- Remove Weblinks
- Remove Punctuation
- Remove Emojis
- Remove Stop Words ("the", "of", etc.)

Tidy

- Each variable is a column
- Each observation is a row
- Each type of observational unit is a table

Bing Sentiment Lexicon Dictionary

- The Bing lexicon is a list of 6,788 English positive, and negative sentiment words.
- The dataset was created primarily to explore the Sentiment associated with product customer reviews.

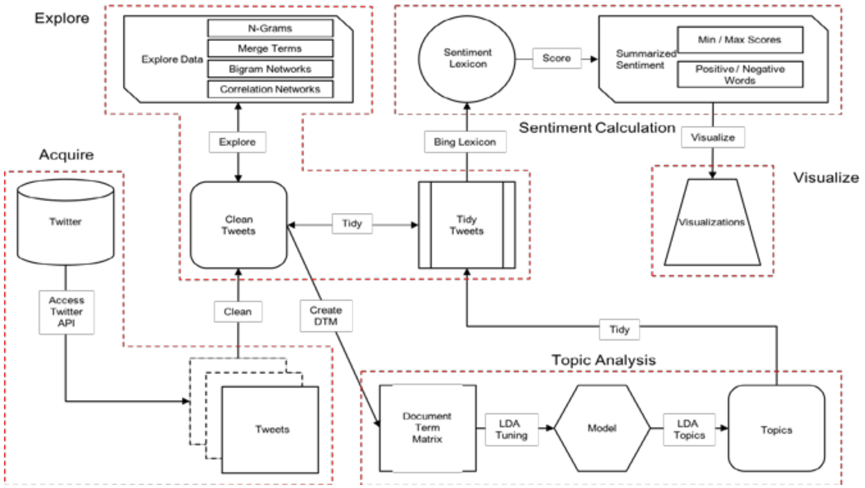
Latent Dirichlet Allocation

LDA is based on two general principles, every document is a mixture of topics, every topic is a mixture of words. Suppose which topic produced every word in a collection was known, except a single word in document D, with word type W. To determine whether this occurrence of W, belongs in topic Z, consider the equation:

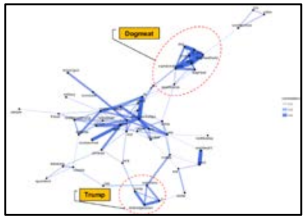
$$P(Z|W,D) = \frac{(\#\,of\ word\ W\ in\ topic\ Z) + \beta_w}{(total\ tokens\ in\ Z) + \beta} * (\#\,words\ in\ D\ that\ belong\ to\ Z + \alpha)$$

CPT Evan Munson
Advisor: LTC Christopher Smith, Ph.D.
Committee Member: Bradley Boehmke, Ph.D.
Department of Operational Sciences (ENS)
Air Force Institute of Technology

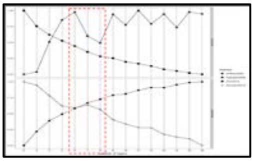
Methodology



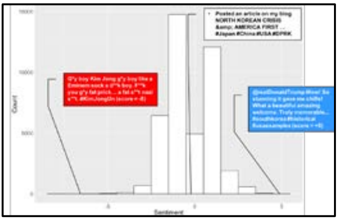
Explore



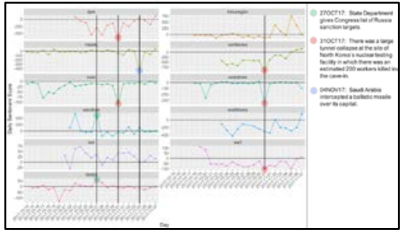
Topic Analysis



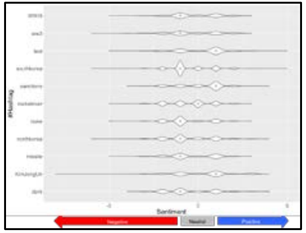
Sentiment Classification



Visualize



Visualize



Tweet Sentiment Score (TSS)

$$TSS = \sum(words_{pos}) + \sum(words_{neg})$$

- Example: "I really love my dog, he is the best friend anyone could ever ask for!"
- Using Bing lexicon: "x xxxxxx **love** xx xxx, xx xx xxx **best** xxxxxx xxxxxx xxxxx xxxx xxx xxx!"

$$TSS = \sum(love\ (+1), best(+1) + \sum(0)$$

$$TSS = +2$$

Future Work

Sentiment Determination:

- Incorporate Emojis
- Account for sarcasm
- Machine Learning classification

Topic Analysis:

- Selecting optimal number of topics
- Tuning LDA algorithm

Conclusions

- Twitter data is free to mine and acquire with, API considerations.
- Methodology provides rapid sentiment analysis compared to traditional Gallup and Pew poll approach.
- Can serve as a time sensitive method to determine Sentiment.
- R package provides sentiment of Twitter data via #hashtags of interest and latent topics uncovered.
- Time series analysis of both hashtag and topic based sentiment analysis provided insight into the publics opinions of global events.
- Not an exact poll, but quickly able to infer knowledge about a subject of interest.

AIR FORCE INSTITUTE OF TECHNOLOGY

Bibliography

1. W. Medhat, A. Hassan, and H. Korashy, “Sentiment analysis algorithms and applications: A survey,” *Ain Shams Engineering Journal*, vol. 5, no. 4, pp. 1093–1113, 2014.

2. V. Basilem and M. Nissim, “Sentiment Analysis.” `http://valeriobasile.github.io/twita/sentix.html`. Accessed: 2017-09-27.

3. S. Mohammad, “NRC Word-Emotion Association Lexicon.” `http://saifmohammad.com/WebPages/NRC-Emotion-Lexicon.htm`, 2016. Accessed:2017-10-02.

4. J. Silge and D. Robinson, *Text Mining with R a Tidy Approach*. Sebastopol: O’Reilly, 1 ed., 2017.

5. M. Nikita, “Select Number ofTtopics for LDA Model,” tech. rep., RPubs, 2016.

6. M. M. Group, “Internet World Users by Language Top 10 Languages.” `https://www.internetworldstats.com/stats7.htm`. Accessed: 2017-09-13.

7. J. Baker and S. Henderson, “Making the Case for Army Data Scientists,” *Army*, vol. 66, pp. 41–43, August 2016.

8. A. Giachanou and F. Crestani, “Like It or Not: A survey of Twitter Sentiment Analysis Methods,” *ACM Computing Surveys*, vol. 49, no. 2, pp. 1–41, 2016.

9. D. M. Boyd and N. B. Ellison, “Social Network Sites: Denition, History, and Scholarship,” *Journal of Computer-Mediated Communication*, vol. 13, no. 1, pp. 210–230, 2007.

10. M. Duggan, D. Page, and S. C. Manager, “Social Media Update 2016.” `http://www.pewinternet.org/2016/11/11/social-media-update-2016/`. Accessed: 2017-05-26.

11. Twitter, “Twitter Usage.” `https://about.twitter.com/company`. Accessed: 2005-07-20.

12. I. L. B. Liu, C. M. K. Cheung, and M. K. O. Lee, “Understanding Twitter Usage: What drive people Continue to Tweet,” in *Proceedings of the Pacific Asia Conference on Information Systems*, pp. 928–939, 2010.

13. H. Wickham and G. Grolemund, *R for Data Science*. Sebastopol: O’Reilly, 1st ed., 2016.

14. D. Newman, “Big Data and the Power of Prediction,” *Forbes*, p. 2. Accessed: 2018-01-18.

15. J. C. Bertot, P. T. Jaeger, and D. Hansen, "The Impact of Polices on Government Social Media Usage: Issues, Challenges, and Recommendations," *Government Information Quarterly*, vol. 29, no. 1, pp. 30–40, 2012.

16. B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," *Foundations and Trends in Information Retrieval*, vol. 2, no. 1-2, pp. 91–231, 2008.

17. A. L. Kavanaugh, E. A. Fox, S. D. Sheetz, S. Yang, L. T. Li, D. J. Shoemaker, A. Natsev, and L. Xie, "Social Media Use by Government: From The Toutine to The Critical," *Government Information Quarterly*, vol. 29, no. 4, pp. 480–491, 2012.

18. B. Costa and J. Boiney, "Social Radar," tech. rep., The MITRE Corporation, 2012.

19. S. Aday, H. Farrell, M. Lynch, J. Sides, and D. Freelon, "New Media and Conflict After the Arab Spring," *United States Institute of Peace*, no. 80, pp. 1–24, 2012.

20. M. Lynch, D. Freelon, and S. Aday, "Syria's Socially Mediated Civil War," *United States Institute of Peace*, no. 91, p. 38, 2014.

21. E. Kouloumpis, T. Wilson, and J. Moore, "Twitter Sentiment Analysis: The Good the Bad and the OMG!," in *Artificial Intelligence*, pp. 538–541, 2011.

22. "Censorship of Twitter." `https://en.wikipedia.org/wiki/Censorship_of_Twitter`. Accessed: 2017-08-01.

23. W. Marcellino, M. Smith, C. Paul, and L. Skrabala, "Monitoring Social Media Lessons for Future Department of Defense Social Media Analysis in Support of Information Operations," tech. rep., 2017.

24. J. M. Boehnert, "Influencing Tomorrow: A Study of Emerging Influence Techniques and Their Relevance to United States Information Operations," tech. rep., Army Command and General Staff College Fort Leavenworth KS, 2015.

25. J. McCain and J. Reed, "National Defense Authorization Act for Fiscal Year 2017, Chairman's Summary," 2017.

26. "Rate Limiting - Twitter Developers." Accessed: 2018-01-16.

27. J. M. Berger and J. Morgan, "The ISIS Twitter Census: Defining and describing the population of ISIS supporters on Twitter," *The Brookings Project on US Relations with the Islamic World*, vol. 3, no. 20, pp. 4–1, 2015.

28. M. A. Hearst, *Search User Interfaces.* New York, NY: Cambridge University Press, 2009.

29. K. Ravi and V. Ravi, "A survey on opinion mining and sentiment analysis: tasks, approaches and applications," *Knowledge-Based Systems*, vol. 89, pp. 14–46, 2015.

30. E. Cambria, D. Das, S. Bandyopadhyay, and A. Feraco, *A practical guide to sentiment analysis*, vol. 5. Springer, 2017.

31. B. Liu, *Sentiment analysis: Mining opinions, sentiments, and emotions.* Cambridge University Press, 2015.

32. G. A. Miller, "WordNet: a lexical database for English," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

33. S. Baccianella, A. Esuli, and F. Sebastiani, "Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining.," in *LREC*, vol. 10, pp. 2200–2204, 2010.

34. S. M. Mohammad and P. D. Turney, "Emotions evoked by common words and phrases: Using Mechanical Turk to create an emotion lexicon," in *Proceedings of the NAACL HLT 2010 workshop on computational approaches to analysis and generation of emotion in text*, pp. 26–34, Association for Computational Linguistics, 2010.

35. S. M. Mohammad and P. D. Turney, "Crowdsourcing a word–emotion association lexicon," *Computational Intelligence*, vol. 29, no. 3, pp. 436–465, 2013.

36. M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 168–177, ACM, 2004.

37. F. Nielsen, "A new ANEW: Evaluation of a Word List for Sentiment Analysis in Microblogs," in *CEUR Workshop Proceedings*, vol. 718, pp. 93–98, 2011.

38. J. Cao, T. Xia, J. Li, Y. Zhang, and S. Tang, "A density-based method for adaptive LDA model selection," *Neurocomputing*, vol. 72, no. 7-9, pp. 1775–1781, 2009.

39. B. A. Frigyik, A. Kapila, and M. R. Gupta, "Introduction to the Dirichlet Distribution and Related Processes," *Department of Electrical Engineering, University of Washignton, UWEETR-2010-0006*, no. 0006, pp. 1–27, 2010.

40. RStudio Team, "Rstudio: Integrated development environment for r." `http://www.rstudio.com/`, 2015.

41. M. Nikita, *ldatuning: Tuning of the Latent Dirichlet Allocation Models Parameters*, 2016. R package version 0.2.0.

42. T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

43. R. Arun, V. Suresh, C. V. Madhavan, and M. N. Murthy, "On finding the natural number of topics with latent dirichlet allocation: Some observations," in *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 391–402, Springer, 2010.

44. R. Deveaud, E. SanJuan, and P. Bellot, "Accurate and effective latent concept modeling for ad hoc information retrieval," *Document numérique*, vol. 17, no. 1, pp. 61–84, 2014.

45. S. Choudhary and B. Adam, "Twitter Analytics Using R Part 1: Extract Tweets." `https://www.credera.com/blog/business-intelligence/twitter-analytics-using-r-part-1-extract-tweets/`. Accessed: 2017-07-3.

46. B. Grün and K. Hornik, "topicmodels: An R package for fitting topic models," *Journal of Statistical Software*, vol. 40, no. 13, pp. 1–30, 2011.

47. M. Z. Asghar, A. Khan, S. Ahmad, M. Qasim, and I. A. Khan, "Lexicon-enhanced sentiment analysis framework using rule-based classification scheme," *PloS one*, vol. 12, no. 2, p. e0171649, 2017.

48. Crowdflower, "Twitter US Airline Sentiment." `https://www.kaggle.com/crowdflower/twitter-airline-sentiment`. Accessed: 2017-10-30.

49. I. Naji, "Twitter Sentiment Analysis Training Corpus Dataset." `http://thinknook.com/twitter-sentiment-analysis-training-corpus-dataset-2012-09-22/`, 2012. Accessed: 2017-08-16.

50. University of Michigan, "UMICH SI650 - Sentiment Classification." `https://www.kaggle.com/c/si650winter11/data`, 2010. Accessed: 2017-11-10.

51. T. Rosentiel, "About Pew Research Center." `http://www.pewresearch.org/about/`. Accessed: 2017-12-19.

52. "Gallup, What We Do." `http://www.gallup.com/corporate/212336/index.aspx`, 2017. Accessed: 2017-12-19.

53. S. Marken and S. Kluch, “The Effects of Probing in Survey Research.” `http://news.gallup.com/opinion/methodology/223286/effects-probing-survey-research.aspx?g{_}source=METHODOLOGYBLOG{&}g{_}medium=topic{&}g{_}campaign=tiles`. Accessed: 2017-12-19.

54. J. Poushter, “Americans hold very negative views of North Korea amid nuclear tensions,” *Pew Research Center*, pp. 1–3, 2017.

55. J. M. Jones, “Pro Football Losing Fans; Other Sports Holding Steady.” `http://news.gallup.com/poll/220562/pro-football-losing-fans-sports-holding-steady.aspx`. Accessed: 2017-12-19.

56. M. Dowle and A. Srinivasan, *data.table: Extension of ‘data.frame‘*, 2017. R package version 1.10.4-3.

57. H. Wickham and J. Bryan, *readxl: Read Excel Files*, 2017. R package version 1.0.0.

58. R Core Team, “R: A language and environment for statistical computing.” `https://www.R-project.org`, 2017.

59. H. Wickham, J. Hester, and R. Francois, *readr: Read Rectangular Text Data*, 2017. R package version 1.1.1.

60. H. Wickham, “The split-apply-combine strategy for data analysis,” *Journal of Statistical Software*, vol. 40, no. 1, pp. 1–29, 2011.

61. H. Wickham, R. Francois, L. Henry, and K. Mller, *dplyr: A Grammar of Data Manipulation*, 2017. R package version 0.7.4.

62. J. Silge and D. Robinson, “tidytext: Text mining and analysis using tidy data principles in r,” *JOSS*, vol. 1, no. 3, 2016.

63. H. Wickham, *stringr: Simple, Consistent Wrappers for Common String Operations*, 2018. R package version 1.3.0.

64. H. Wickham, *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York, 2009.

65. G. Grolemund and H. Wickham, “Dates and times made easy with lubridate,” *Journal of Statistical Software*, vol. 40, no. 3, pp. 1–25, 2011.

66. T. L. Pedersen, “ggraph: An implementation of grammar of graphics for graphs and networks,” 2018. R package version 1.0.1.

67. G. Csardi and T. Nepusz, “The igraph software package for complex network research,” *InterJournal*, vol. Complex Systems, p. 1695, 2006.

68. D. Robinson, *widyr: Widen, Process, then Re-Tidy Data*, 2017. R package version 0.1.0.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704–0188*

1. REPORT DATE *(DD–MM–YYYY)* 22-03-2018	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* SEP 2016 - MAR 2018

4. TITLE AND SUBTITLE Sentiment Analysis of Twitter Data	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Munson, Evan, L. CPT, U.S. Army	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT-ENS-MS-18-M-148
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Intentionally Left Blank	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The rapid expansion and acceptance of social media has opened doors into users opinions and perceptions that were never as accessible as they are with todays prevalence of mobile technology. Harvested data, analyzed for opinions and sentiment can provide powerful insight into a population. This research utilizes Twitter data due to its widespread global use, in order to examine the sentiment associated with users tweets. An approach utilizing Twitter #hashtags and Latent Dirichlet Allocation topic modeling were utilized to differentiate between tweet topics. A lexicographical dictionary was then utilized to classify sentiment. This method provides a framework for an analyst to ingest Twitter data, conduct an analysis and provide insight into the sentiment contained within the data.

15. SUBJECT TERMS

Twitter, Sentiment Analysis, Lexicographical, Bing Dictionary, Topic Modeling, Latent Dirichlet Allocation, Army

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			LTC C. M. Smith, Ph.D., AFIT/ENS
U	U	U	UU	132	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255-3636, x4318; christopher.smith@afit.edu